

TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky a mezioborových inženýrských studií

DIPLOMOVÁ PRÁCE

Regulátor s proměnnou strukturou

Controler with variable structure

ANOTACE

Diplomová práce se zabývá řízením systémů s parametry proměnnými v čase při použití přepínaného regulátoru s paralelními modely. V algoritmu přepínaného regulátoru je využito vlastností upraveného Luenbergerova estimátoru, který estimuje stav soustavy bezchybně v případě vstupu dané poruchy do regulovaného obvodu.

Byl navržen a ověřen nový algoritmus, který zapojuje do regulačního obvodu různě naladěné regulátory. Tyto regulátory jsou naladěny pro nominální regulovaný systém a pro několik zvolených nenominálních systémů. Pro každý regulovaný systém je naladěno několik regulátorů pro poruchy vstupující v různých místech regulačního obvodu. Vzhledem k jednomu neproměnnému regulátoru byla dosažena vyšší robustnost a též kvalita regulace. Navržený algoritmus byl testován při deterministických a stochastických poruchách a při zvolených změnách parametrů regulovaného systému.

ABSTRACT

This thesis deals with the control of systems with time varying parameters using switched controller with parallel models. The algorithm of switched controller utilizes the properties of modified Luenberger estimator. The modification of the Luenberger estimator is made in such a way, that each estimator is designed for a particular disturbance type and its estimation remains correct as long as the system is affected only by this particular disturbance.

The new algorithm designed and tested in this thesis uses switching logic to select among a set of differently tuned controllers. The controllers are tuned for nominal process model as well as for several non-nominal models. Further, there are several controllers associated with each model and each of them is tuned to reject disturbance affecting different part of the control loop. The testing of this algorithm has included responses to deterministic and stochastic disturbances as well as sensitivity to parameter changes. It has been shown that control performance and robustness of this switched multi-model algorithm is increased in comparison with one fixed structure controller.

Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 o právu autorském, zejména § 60 (školní dílo).

Beru na vědomí, že TUL má právo na uzavření licenční smlouvy o užití mé DP a prohlašuji, že souhlasím s případným užitím mé diplomové práce (prodej, zapůjčení apod.).

Jsem si vědom toho, že užít své diplomové práce či poskytnout licenci k jejímu využití mohu jen se souhlasem TUL, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce a konzultantem.

Datum:

Podpis:

OBSAH:

MODEL SYSTÉMU	1
VNĚJŠÍ A VNITŘNÍ POPIS	1
<i>Vnější popis</i>	1
<i>Vnitřní popis</i>	3
VZTAH MEZI VNĚJŠÍM A VNITŘNÍM POPISEM.....	5
ŘÍZENÍ DYNAMICKÉHO SYSTÉMU.....	6
DISKRÉTNÍ REGULAČNÍ OBVODY	6
STAVOVÝ REGULÁTOR	8
NÁVRH STAVOVÉHO REGULÁTORU PODLE KVADRATICKÉHO KRITÉRIA	9
VÝPOČET STAVOVÝCH REGULÁTORŮ PŘI POUŽITÍ MATLABU	9
PORUCHY	10
CHYBA MODELU	10
STRUKTURA DISKRÉTNÍCH MODELŮ PORUCH.....	11
<i>ARX (Auto-Regressive with eXogenous variable)</i>	11
<i>OE (Output Error)</i>	12
<i>ARMAX (Auto-Regressive Moving Average with eXogenous variable),</i>	12
<i>IN (Input error)</i>	13
ESTIMACE	13
DETERMINISTICKÝ ESTIMÁTOR STAVU	14
ESTIMÁTOR PRO BEZCHYBNÝ ODHAD VEKTORU STAVU	16
ČASOVÁ AFINITA	18
REGULÁTOR S PROMĚNNOU STRUKTUROU	19
SIMULAČNÍ PROGRAM – MATLAB	21
FUNKCE V MATLABU	21
SIMULINK.....	22
STANDARDNÍ KNIHOVNY	22
S-FUNKCE	22
REALIZACE SIMULAČNÍHO MODELU.....	23
PRINCIP PŘEPÍNÁNÍ MEZI REGULÁTORY	26
ALGORITMUS PŘEPÍNAČÍ FUNKCE	26
POPIS OVLÁDÁNÍ SIMULAČNÍHO PROGRAMU	30
OPTIMALIZACE MATICE Q	30
VLASTNÍ MINIMALIZAČNÍ FUNKCE – M-FILE	32
POPIS PRŮBĚHU MINIMALIZACE POMOCÍ FUNKCE <i>FMINSEARCH</i>	32
OVĚŘENÍ VLASTNOSTÍ	33
TESTOVÁNÍ PŘI VSTUPU DETERMINISTICKÝCH PORUCH - SOUSTAVA 1	33
<i>Porucha IN</i>	34
<i>Porucha ARX</i>	35
<i>Porucha ARMAX</i>	36
<i>Porucha OUT</i>	37
TESTOVÁNÍ PŘI VSTUPU DETERMINISTICKÝCH PORUCH - SOUSTAVA 2	38
<i>Porucha IN</i>	39
<i>Porucha ARX</i>	40
<i>Porucha ARMAX</i>	41
<i>Porucha OUT</i>	42
TESTOVÁNÍ S „PARAZITNÍ KAPACITOU“ I.....	43
<i>Porucha IN</i>	44
<i>Porucha ARX</i>	44
<i>Porucha ARMAX</i>	45
<i>Porucha OUT</i>	45
TESTOVÁNÍ S „PARAZITNÍ KAPACITOU“ II.....	46

<i>Porucha IN</i>	46
<i>Porucha ARX</i>	47
<i>Porucha ARMAX</i>	47
<i>Porucha OUT</i>	48
TESTOVÁNÍ PŘI VSTUPU STOCHASTICKÝCH PORUCH	48
<i>Porucha IN</i>	48
<i>Porucha ARX</i>	49
<i>Porucha ARMAX</i>	49
<i>Porucha OUT</i>	49
HODNOCENÍ	50
ZÁVĚR	50
PŘÍLOHA A	51
SEZNAM POUŽITÉ LITERATURY	52

Úvod

Je velice pravděpodobné, že se během chodu navrhovaného zařízení budou částečně měnit jeho parametry vlivem okolního prostředí, a zároveň se při působení jakékoli energie každá součástka více či méně opotřebovává. Proto je potřeba s tímto počítat již při návrhu a přizpůsobit regulační členy tak, aby byly schopny dobře řídit soustavu i při nenominálních podmínkách.

Jednou možností je návrh robustního regulátoru. Jeho výhoda spočívá v tom, že řídicí člen nemusí být inteligentním prvkem a regulátor reaguje okamžitě s maximální účinností, která je však omezená vzhledem k nastavení jeho parametrů. Regulátor musí být navržen tak, aby byl schopen stabilně řídit systém, aniž by se měnila jeho struktura či parametry. Z tohoto předpokladu však také vyplývá jeho nevýhoda, a to, že kvalita regulace v podstatě závisí na jeho robustnosti. Neboli čím robustnější regulátor navrhnu, tím horší bude kvalita regulace.

Druhou možností jsou adaptivní regulátory, které jsou poměrně náročné na hardware regulačního členu, neboť v průběhu regulace přepočítávají své parametry podle aktuálního stavu regulované soustavy, resp. výstupní veličiny. A proto nejsou tyto regulátory schopny optimálně reagovat na rychlé změny vstupujících poruch.

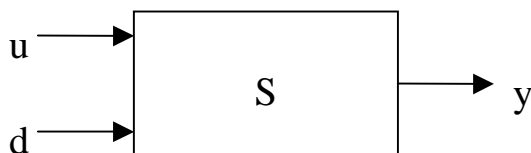
Třetí možností jsou regulátory s proměnnou strukturou, které se vyznačují poměrně velkou robustností při zachování kvality regulace a jejich nároky na výkon regulačního členu by měly být menší než u adaptivních regulátorů. Vše však závisí na zvolení vhodné struktury a na algoritmu přepínání, což je náplní této diplomové práce.

1. Model systému

Dynamické vlastnosti skutečných průmyslových zařízení se vyjadřují pomocí modelu systému, který sestavujeme na základě fyzikálních vlastností a vazeb, takto vznikne fyzikální model, který poté upřesňujeme pomocí měření vstupů a výstupů na reálné soustavě.

Tímto způsobem však není možné popsat identifikovanou soustavu úplně dokonale přesně tak, aby byl model dále použitelný. Zavádíme proto některá zjednodušení.

Obecné schéma modelu systému je zobrazeno na Obr.1.1.



Obr.1.1 Model systému

Kde u je soubor signálů vstupujících do systému, které můžeme ovlivňovat. Dále do systému vstupuje množina signálů, které není možno ovlivnit a označujeme je jako poruchy – d . Na systému potom můžeme měřit výstupní veličinu, označujeme y , která vzniká jako suma reakcí systému na vstupní signály.

Modely je možno rozdělit podle toho jak věrně popisují(znázorňují) daný systém. Při tvorbě modelu potom hovoříme o systémech rozloženými parametry, jestliže uvažujeme to, že hmoty a media jsou rozložena v prostoru a v každém bodě mohou mít různé parametry. Jejich vlastnosti v každém bodě popisují parciální diferenciální rovnice. Druhým typem jsou systémy s koncentrovanými parametry, u kterých považujeme celou hmotu či medium za hmotný bod.

Vnější a vnitřní popis

Vnější popis systému získáme pokud budeme vyjadřovat relace pouze mezi vstupními a výstupními veličinami. Vnější popis nám tedy nic neříká o vnitřních stavech. Tyto relace mezi vstupem a výstupem mohou být vyjádřeny:

obrazovým přenosem
diferenciální rovnicí
impulsní přechodovou (váhovou) funkcí
frekvenční charakteristikou
přechodovou charakteristikou

a) diferenciální rovnice

Lineární dynamický systém je možno popsat lineární diferenciální rovnicí řádu n , který je roven řádu systému. Lineární časově neproměnný systém se označuje zkratkou LTI (Linear Time Invariant = lineární časově nezávislé).

$$a_n y^{(n)}(t) + a_{n-1} y^{(n-1)}(t) + \dots + a_0 y(t) = b_m u^{(m)}(t) + \dots + b_0 u(t) \quad (1)$$

kde a_i, b_i jsou konstantní koeficienty

$y(t)$ je výstupní veličina

$u(t)$ je vstupní veličina

b) obrazový přenos – laplaceova transformace

- je roven laplaceovu obrazu výstupní veličiny ku laplaceově obrazu vstupní veličiny, při nulových počátečních podmínkách zleva.
- je roven laplaceovu obrazu impulsní přechodové (váhové) funkce
- získáme pokud aplikujeme laplaceovy transformační vztahy na diferenciální rovnici (1) soustavy.

$$y(t) = Y(s) \quad (2)$$

$$y^{(n)}(t) = s^n Y(s) - s^{n-1}y(0) - s^{n-2}y^{(1)}(0) - \dots - y^{(n-1)}(0)$$

,kde horní koeficienty v závorkách značí řád derivace a koeficienty bez závorek řád mocniny. Diferenciální rovnice v laplaceově obrazu potom vypadá následovně.

$$(a_n s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0) Y(s) = (b_m s^m + b_{m-1} s^{m-1} + \dots + b_1 s + b_0) U(s) \quad (3)$$

,kde $Y(s)$ a $U(s)$ jsou laplaceovy obrazy veličin $y(t)$ a $u(t)$, s je komplexní proměnná. Rovnici (3) můžeme zapsat:

$$A(s) Y(s) = B(s) U(s) \quad (4)$$

$$\frac{Y(s)}{U(s)} = \frac{B(s)}{A(s)} = F(s) \quad (5)$$

,kde podíl polynomů $\frac{B(s)}{A(s)} = F(s)$ se nazývá obrazový přenos. Potom kořeny polynomu

v čitateli se nazývají nuly a kořeny polynomu ve jmenovateli póly soustavy. Obrazový přenos lze pak vyjádřit jako podíl součinů kořenových činitelů.

$$F(s) = \frac{b_m (s - n_1)(s - n_2) \dots (s - n_m)}{a_n (s - p_1)(s - p_2) \dots (s - p_n)} \quad (6)$$

,kde n_i jsou nuly a p_i póly

c) obrazový přenos – Z-transformace

Z-transformace slouží k popisu diskrétně pracujícího systému, jehož vstupy a výstupy jsou reprezentovány posloupnostmi čísel.

V současné době, kdy se v regulační technice začíná čím dál tím více jako regulačních členů používat počítače a mikropočítače, je třeba signál, který vystupuje ze soustavy, tzv. nevzorkovat pomocí A/Č (analogovo/číslíkového) převodníku. A naopak signál, který vystupuje z diskrétního regulačního prvku je potřeba převést do analogové podoby, což se provádí Č/A (číslíkově/analogovým) převodníkem.

V časové oblasti tomuto popisu systému odpovídá diferenční rovnice:

$$y(j) + a_2 y(j-1) + \dots + a_n y(j-n) = b_2 u(j-1) + b_3 u(j-2) + \dots + b_n u(j-n) \quad (7)$$

,která popisuje systém n -tého řádu. Tato rovnice potom v z -transformaci vypadá následovně:

$$(1 + a_2 z^{-1} + a_3 z^{-2} + \dots + a_{n+1} z^{-n}) Y(z^{-1}) = (b_2 z^{-1} + b_3 z^{-2} + \dots + b_{n+1} z^{-n}) U(z^{-1}) \quad (8)$$

Rovnici (8) dále můžeme zapsat ve tvaru:

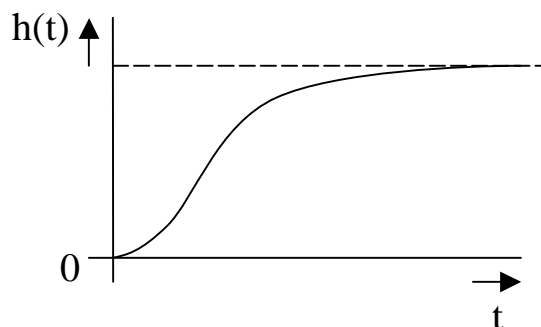
$$A(z^{-1}) Y(z^{-1}) = B(z^{-1}) U(z^{-1})$$

$$\frac{Y(z^{-1})}{U(z^{-1})} = \frac{B(z^{-1})}{A(z^{-1})} = F(z^{-1}) \quad (9)$$

Podíl polynomů $\frac{B(z^{-1})}{A(z^{-1})} = F(z^{-1})$ nazýváme diskrétní přenos systému.

d) přechodová charakteristika

- je grafickým vyjádřením přechodové funkce. Je to odezva systému na jednotkový skok při nulových počátečních podmínkách. Tuto charakteristiku měříme tak, že necháme systém ustálit, poté přivedeme na vstup jednotkový skok a opět čekáme až do ustálení. Pro zpřesnění můžeme provést totéž měření několikrát a výsledky zprůměrnovat.
- hodnota ustálení této charakteristiky je rovna zesílení systému.



Obr.1.2 Přechodová charakteristika soustavy 2. řádu

Vnitřní popis je popis relací mezi vstupní veličinou $u(t)$, vnitřním stavem systému $x(t)$ a výstupní veličinou $y(t)$. Veličiny popisující stav vnitřním stav systému nazýváme stavové veličiny. Matematickým vyjádřením těchto relací jsou stavové rovnice.

Obecně lze dynamický systém popsat nelineární vektorovou stavovou rovnicí ve tvaru:

$$\dot{x}(t) = f(x, u, t) \quad (10)$$

$$y(t) = g(x, u, t) \quad (11)$$

,kde $x(t)$. . je n-rozměrný stavový vektor

$y(t)$. . je výstupní veličina systému

f je n-rozměrná nelineární funkce

g je skalární funkce

a) **spojitý stavový popis**

Je-li dynamický systém lineární časově nezávislý (LTI), pak platí :

$$\begin{aligned} \dot{x}(t) &= A \cdot x(t) + B \cdot u(t) \\ y(t) &= C \cdot x(t) + Du(t) \end{aligned} \quad (12)$$

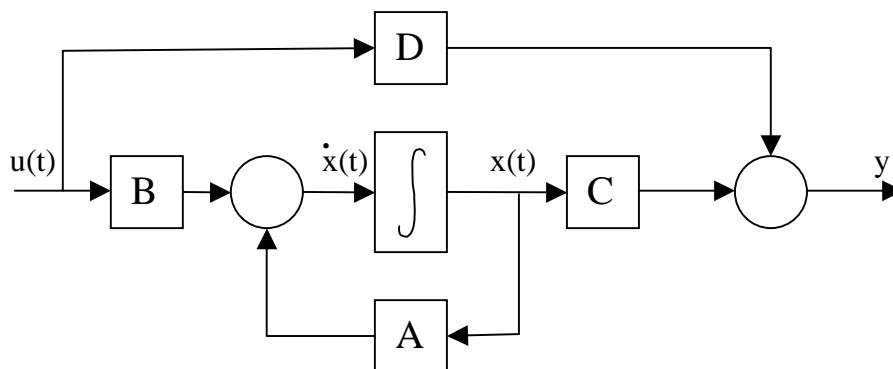
,kde A je matice systému o rozměru $n \times n$

B je matice buzení a má rozměr $n \times 1$

C je výstupní matice rozměru $1 \times n$

D je koeficient převodu

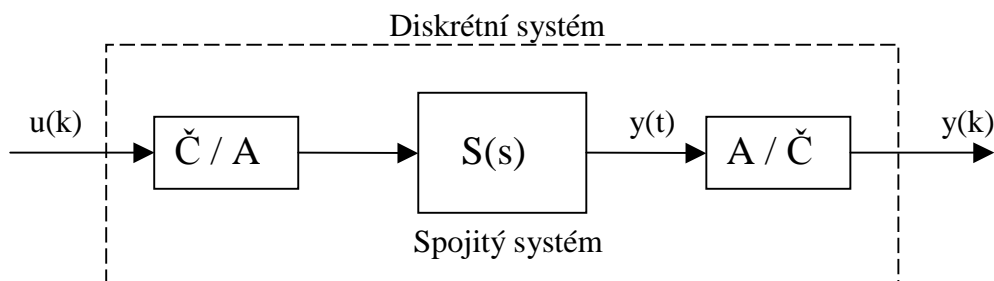
Pokud provádíme matematicko-fyzikální analýzu dynamických systémů při aplikacích makroskopických bilancí hmoty a energie dostáváme většinou soustavu rovnic prvního řádu, tedy přímo stavový popis systému.



Obr.1.3 Blokové schéma zapojení

b) diskrétní stavový popis

Připojením převodníků na vstup a výstup spojitého systému se vytvoříme diskrétní systém. Přičemž na vstupu systému je číslicově-analogový převodník (Č/A převodník) a na výstupu převodník analogovo-číslcový (A/Č převodník). Znázorněno na obr.1.4.



Obr.1.4

Převodník A/Č měří periodicky výstupní veličiny $y(t)$ spojitého systému. A/Č převodník se také nazývá vzorkovací člen. Na výstupu tohoto převodníku je tedy posloupnost $y(k)$. Na vstupu převodníku Č/A je diskrétní posloupnost $u(k)$ a na jeho výstupu je analogový signál $u(t)$, který vstupuje do spojitého systému. Převodník Č/A provádí tvarování vstupní posloupnosti do spojitého schodového signálu, nazývá se tvarovací člen.

Stavové rovnice pro diskrétní systém vypadají následovně

$$\begin{aligned} x(k+1) &= Mx(k) + Nu(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned} \quad (13)$$

,kde je M matice diskrétních koeficientů $n \times n$

N matice buzení $n \times p$

$x(k)$ vektor disk.stavových veličin $n \times 1$

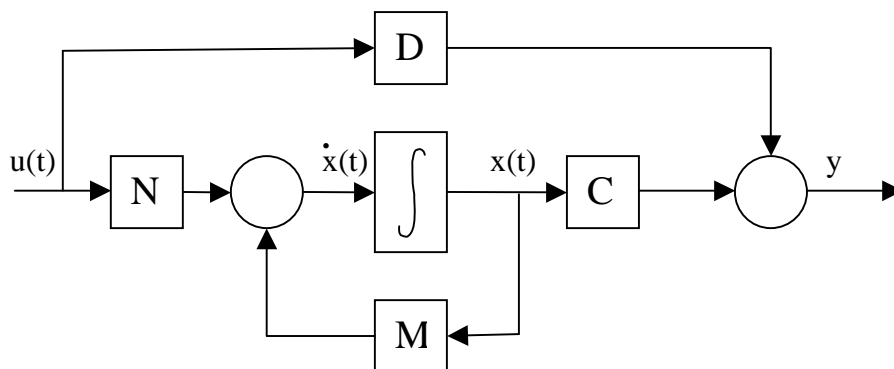
$u(k)$ vektor diskrétního buzení $p \times 1$

C matice výstupu $m \times n$

D matice převodu $m \times p$

p počet budících veličin

$y(k)$ výstupní veličina



Obr.1.5 Blokové schéma zapojení diskretních matic – stavový popis

Vztah mezi vnějším a vnitřním popisem

Stavový popis systému je tvořen soustavou diferenčních rovnic prvního řádu. Proměnné těchto diferenčních rovnic tvoří souřadný systém stavového prostoru. Stavové proměnné systému je možno využít při řízení a regulaci. Přejít ze spojitého obrazového přenosu na spojitý stavový popis lze provést různými metodami, např. metodou snižování řádu derivace nebo metodou postupných integrací. Metoda snižování řádu derivace vede na strukturu stavového popisu, který se nazývá normální forma říditelnosti. Metoda postupných integrací vede na normální formu pozorovatelnosti. Strukturou obdobné stavové popisy systémů se též používají pro diskretní systémy. V projektu byla použita pouze normální forma pozorovatelnosti diskretních systémů, kde struktura vektoru stavu $\mathbf{x}(k)$, matice \mathbf{M} , matice buzení \mathbf{N} a matice výstupu \mathbf{C} je

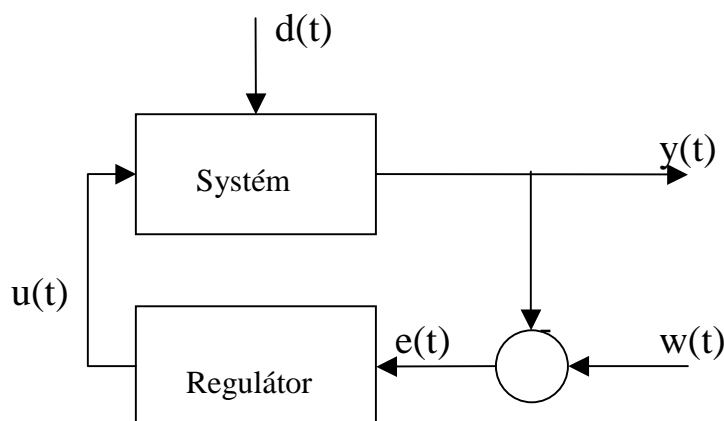
$$\mathbf{x}(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ \vdots \\ x_{n-1}(k) \\ x_n(k) \end{bmatrix}; \mathbf{M} = \begin{bmatrix} -a_{n+1} & 1 & 0 & \cdots & 0 \\ -a_n & 0 & 1 & \cdots & 0 \\ -a_{n-1} & 0 & 0 & \cdots & 0 \\ \vdots & & & & \\ -a_3 & 0 & 0 & \cdots & 1 \\ -a_2 & 0 & 0 & \cdots & 0 \end{bmatrix}; \mathbf{N} = \begin{bmatrix} b_{n+1} \\ b_n \\ b_{n-1} \\ \vdots \\ b_3 \\ b_2 \end{bmatrix}; \mathbf{C} = [1, 0, 0, \dots, 0]; \mathbf{D} = [0] \quad (14)$$

Matici převodu \mathbf{D} v našem případě uvažujeme nulovou.

2. Řízení dynamického systému

Můžeme rozdělit do dvou kategorií a to na zpětnovazební, nebo-li regulace, a na přímovazební, nebo-li ovládání.

- Přímovazební řízení – řídíme systém na základě znalosti jeho chování a velikosti vstupní veličiny. Při tomto druhu řízení neporovnáváme výstupní veličinu s žádanou hodnotou výstupu.
- Zpětnovazební řízení - pomocí regulátoru se snažíme udržet výstupní veličinu stejně velkou jako je hodnota žádané veličiny. K tomu účelu porovnáváme výstup se žádanou veličinou a výsledek zavádíme přes regulátor zpět na vstup regulovaného systému.



Obr.2.1 Obecné regulační schéma

,kde $w(t)$ je žádaná hodnota

$y(t)$ je výstupní hodnota

$e(t)$ je regulační odchylka, kterou můžeme vyjádřit jako:

$$e(t) = w(t) - y(t) \quad (15)$$

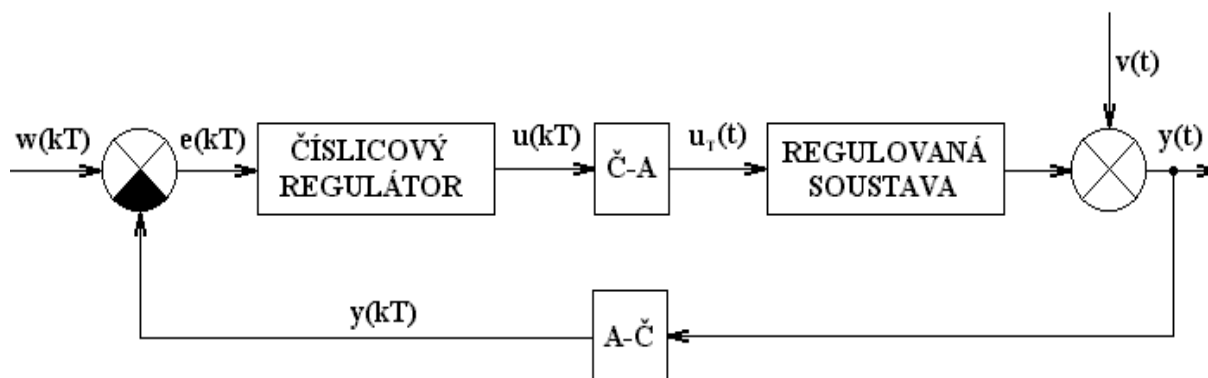
$u(t)$ označujeme jako akční veličinu

$d(t)$ je poruchová veličina

Diskrétní regulační obvody

Diskrétní regulační obvod je takový obvod, ve kterém alespoň jedna veličina má tvar posloupnosti diskretních hodnot vytvářených např. v časově ekvidistantních okamžicích. Obvykle si představujeme, že diskretní regulační obvod využívá k výpočtu akční veličiny číslicový počítač, což však nemusí platit vždy.

Diskrétní regulační obvod, viz Obr.2.2, lze znázornit blokovým schématem skládajícím se ze spojitě pracujících členů, ze vzorkovacích členů, z tvarovacích členů, nespojitě (diskrétně) pracujících členů, analogově-číslicových převodníků, z číslicově-analogových převodníků a popřípadě i z paměťových členů.



Obr.2.2 Obecné blokové schéma lineárního diskrétního regulačního obvodu

,kde u_r - tvarovaná akční veličina

k, kT - diskrétní čas ($k = 0, 1, 2, \dots$)

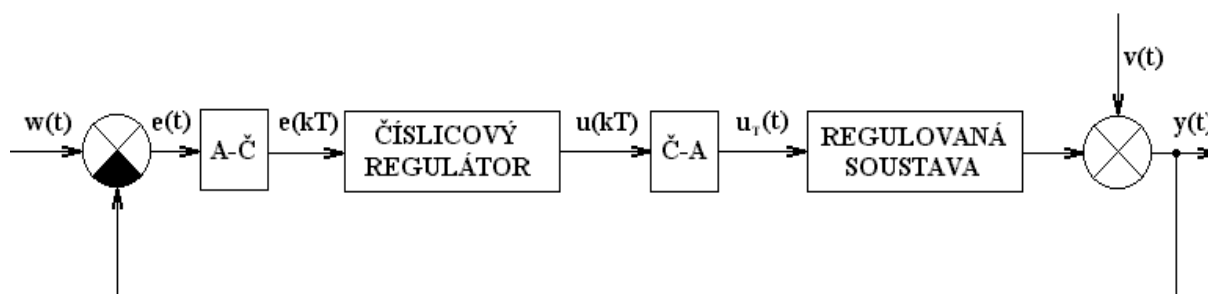
T - vzorkovací perioda

A/Č - analogově číslicový převodník

Č/A - číslicově analogový převodník

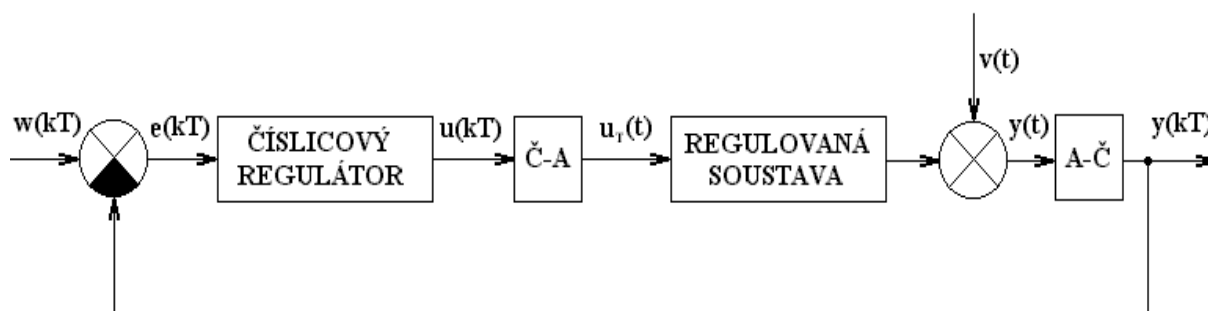
Regulovaná soustava je vždy spojitá. Při řešení syntézy diskrétních regulačních obvodů se obecně používají dva postupy:

a) předsunutí A-Č – převodníky se uvažují jako součást regulátoru, pak pracujeme s kvazispojitým regulátorem, viz Obr.2.3. Tento postup je možný při malé vzorkovací periodě. Až 80% číslicových regulačních obvodů můžeme převést na analogové regulační obvody.



Obr.2.3 Blokové schéma lineárního regulačního obvodu s kvazianalogovým regulátorem

b) podsunutí A-Č – převodníky se přidávají k regulované soustavě, kterou považujeme jako kvazidiskrétní, viz Obr.2.4. Jestliže dáme A/Č za regulovanou soustavu, považujeme obvod za diskrétní. Tento postup se hlavně využívá při velkých vzorkovacích periodách. Používáme Z-transformaci, která je dosti složitá. Nedochází ke zkreslení informace.



Obr.2.4 Blokové schéma lineární diskrétního regulačního obvodu

Stavový regulátor

Oproti běžným PID (proporcionálně-integračně-derivačním) regulátorům, má stavový regulátor mnohem jednodušší strukturu. Je tvořen pouze jednou maticí koeficientů, pomocí které se násobí každý stav systému a suma těchto součinů tvoří akční veličinu.

Regulátor mohu navrhnout podle několika různých kriterií, jsou jimi:

- minimální lineární regulační plocha
- minimální kvadratická regulační plocha
- metoda optimálního modulu

Pro seřízení regulátoru v této diplomové práci byla vybrána metoda minimální kvadratické regulační plochy.

Uvažujme nejprve případ, že je na systému možné měřit všechny jeho stavové veličiny – složky vektoru $x(k)$. Stavovou regulaci reprezentuje rovnice

$$u(k) = K \cdot x(k), \quad (16)$$

kde K je matice zpětnovazebních koeficientů.

Stavovým regulátorem je realizována proporcionální zpětná vazba, která není schopna odstranit trvalou regulační odchylku. Z tohoto důvodu je zapotřebí do regulačního obvodu zapojit přídatný integrátor.

V diplomové práci je diskutován regulátor s proměnnou strukturou ve kterém je integrátor zapojen před systém, z rovnice regulátoru vystupují přírůstky akční veličiny, (16) upravíme do tvaru

$$\Delta u(k) = K \cdot x(k). \quad (17)$$

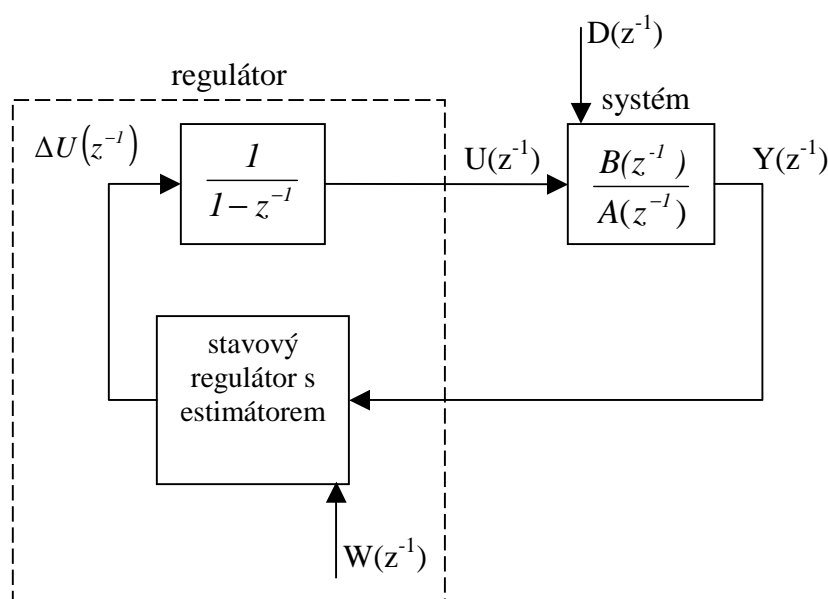
Přírůstky akční veličiny se v řídicím členu - počítači integrují - sčítají,

$$u(j) = \sum_{k=1}^j \Delta u(k). \quad (18)$$

Veličina $u(j)$ přes Č/A převodník vstupuje do systému.

Přenos členu (18) v Z transformaci je

$$\frac{U(z^{-1})}{\Delta U(z^{-1})} = \frac{1}{1 - z^{-1}} \quad (19)$$



Obr.2.5 Regulační obvod s integrátorem na vstupu systému

Do modelu systému – estimátoru stavu je zapojen integrátor na výstupu. Tato struktura estimátoru umožňuje jednoduše zadávat žádanou hodnotu regulované – výstupní veličiny systému.

Návrh stavového regulátoru podle kvadratického kritéria

U kritéria kvadratické regulační plochy je s je definováno jako $s(t) = e(t) - e(\infty)$. Kriterium je definováno integrálem, v němž je druhá mocnina v integrandu nahrazena komplexní konvolucí v Laplaceově obrazu.

$$J_k = \int_0^{\infty} s^2(t) dt = \lim_{p \rightarrow 0} \left[\frac{1}{2\pi j} \cdot \int_{c-j\infty}^{c+j\infty} S(p-q) S(q) dq \right] \quad (20)$$

Hodnota integrálu se počítá z reziduové věty a vede na vzorce, často publikované v literatuře, např. [Kotek, Štecha 78]. Kriterium pokutuje silně velké odchylky v porovnání s odchylkami malými a v důsledku toho vede na příliš kmitavé průběhy. Z tohoto důvodu byla odvozena tzv. *metoda zobecněné kvadratické plochy* s kriteriem

$$J'_k = \int_0^{\infty} \sum_{i=0}^{n-1} q_i \cdot \left(s^{(i)} \right)^2 dt \quad (21)$$

,která váží též čtverce derivací sledovaného průběhu. Při vhodné volbě koeficientů q_i může být průběh správně zatlumen.

Kvadratická kritéria jsou východiskem mnoha metod v moderní teorii optimálního řízení.

Vycházejí ze stavového popisu soustavy a z kritéria

$$J_k = \int_0^{\infty} \left(\mathbf{x}^T(t) \cdot \mathbf{Q} \cdot \mathbf{x}(t) + \mathbf{u}^T(t) \cdot \mathbf{R} \cdot \mathbf{u}(t) \right) dt \quad (22)$$

,kde $\mathbf{x}(t)$ a $\mathbf{u}(t)$ jsou vektory stavových, resp. vstupních proměnných soustavy, popsané stavovým popisem. Matice váhových koeficientů \mathbf{Q} a \mathbf{R} mohou za jistých podmínek mít fyzikální interpretaci. Matice \mathbf{R} je pro systémy s jedním vstupem a výstupem pouze koeficient.

Výpočet stavových regulátorů při použití MATLABu

Pro výpočet stavového regulátoru podle kritéria minimální kvadratické regulační plochy je možno použít funkce *dlqr*. Tato funkce je definována následovně

$$[K, S, E] = \text{dlqr}(M, N, Q, R)$$

K - optimální diskretní stavový regulátor minimalizující kritérium

S - řešení Riccatiho rovnice

E - vlastní čísla uzavřeného obvodu

M - matice systému

N - matice buzení

Q - váhová matice stavových veličin v kvadratickém kritériu

R - váhová matice (pro SISO systémy - koeficient) u akční veličiny v kvadratickém kritériu

\mathbf{Q} a \mathbf{R} jsou symetrické pozitivně semidefinitní matice. Jsou navrženy jako diagonální matice s kladnými prvky.

3. Poruchy

V této kapitole bude vysvětlen vstup jednotlivých typů poruch do soustavy. Je to důležité kvůli následné diskuzi estimátoru vyvinutého na TU v Liberci.

Pro účely simulace zavádíme několik modelů poruch, které mohou vstoupit na vstup soustavy, na výstup nebo přímo do soustavy v průběhu regulace.

Existují dva základní typy poruch, které mohou soustavu ovlivňovat

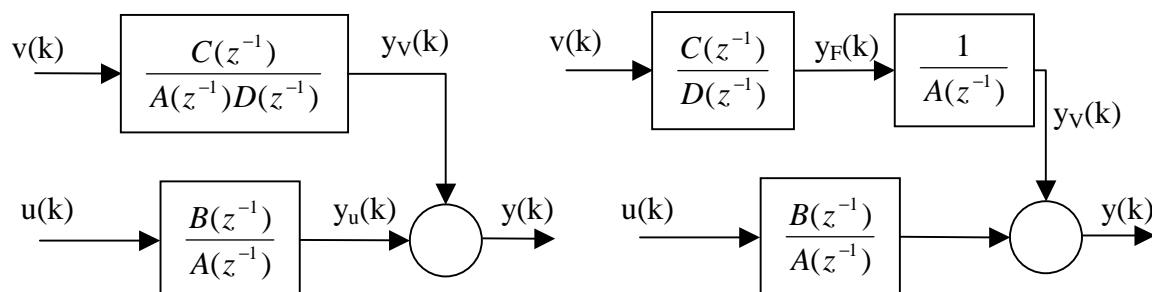
a) Stochastické

b) Deterministické

a) Jsou to náhodné signály, které nelze analyticky popsat, každá realizace je neopakovatelná. Popisují se pomocí střední hodnoty, rozptylu, autokorelační funkce a výkonové spektrální hustoty.

b) Jsou signály, které můžeme analyticky popsat. Mezi nejznámější patří jednotkový skok, rampa, sinus atd. Tyto signály můžeme označovat také jako testovací.

Je-li možno dynamický systém popsat stavovými rovnicemi, pak budou jmenovatele přenosů poruch obsahovat stejný polynom jako jmenovatel přenosu soustavy.



Obr.3.1 Schéma modelu s deterministickými i stochastickými poruchami

Model má dva vstupy (viz Obr.3.1), jeden pro determinovaný vstup $u(k)$ a druhý pro stochastický signál – bílý šum $v(k)$ procházející přes lineární filtr $G_F(z^{-1}) = C(z^{-1}) / D(z^{-1})$, jehož výstupem je stacionární stochastický signál y_F .

$$\begin{aligned}
 y(k) &= y_u(k) + y_v(k) = \frac{B(z^{-1})}{A(z^{-1})} u(k) + \frac{C(z^{-1})}{D(z^{-1})A(z^{-1})} v(k) = \\
 &= \frac{B(z^{-1})}{A(z^{-1})} u(k) + G_F(z^{-1}) \frac{1}{A(z^{-1})} v(k)
 \end{aligned} \tag{23}$$

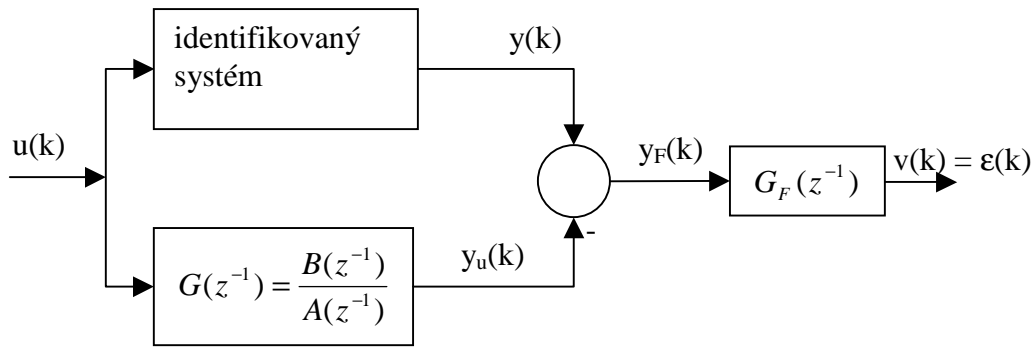
Vynásobíme-li rovnici (23) polynomem $A(z^{-1})$ dostaneme rovnici

$$y(k) \cdot A(z^{-1}) - B(z^{-1})u(k) = \frac{C(z^{-1})}{D(z^{-1})} v(k) = G_F(z^{-1})v(k) \tag{24}$$

Chyba modelu

Mějme na paměti, že model má sloužit k identifikaci dynamických systémů, měla by nás tedy zajímat i chyba modelu.

Jedna z možných struktur je na Obr.3.2. Pro získání chyby, budeme uvažovat ideální stav, kdy polynomy $A(z^{-1})$, $B(z^{-1})$, $C(z^{-1})$, $D(z^{-1})$ modelu jsou shodné s polynomy skutečného systému.



Obr.3.2

Nejdříve určíme rozdíl $y(k) - y_u(k) = \Delta(k)$.

Z předpoklad, že o odhadech parametrů platí $\Delta(k) = y(k) - y_u(k) = y_F(k)$.

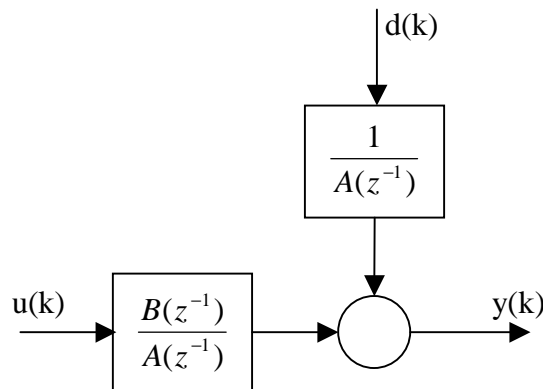
Chyba výstupu modelu je tedy rovna

$$v(k) = G_F^{-1}(z^{-1})\Delta(k) = G_F^{-1}(z^{-1}) \left[y(k) - \frac{B(z^{-1})}{A(z^{-1})}u(k) \right] \quad (25)$$

Struktura diskrétních modelů poruch

Při identifikaci používáme různé struktury modelů, které vycházejí z rovnosti (25) a podle šumového signálu volíme $G_F(z^{-1})$

- a) **ARX** (Auto-Regressive with eXogenous variable) – pokud zvolíme $G_F(z^{-1}) = 1$, pak dostaneme model poruchy tak, jak je na Obr.3.3.



Obr.3.3 Model ARX

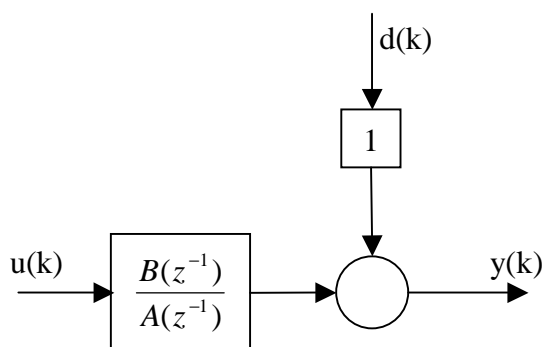
Výstup tohoto modelu vyjádříme takto

$$y(k) = \frac{B(z^{-1})}{A(z^{-1})}u(k) + \frac{1}{A(z^{-1})}v(k) \quad (26)$$

a diferenční rovnice modelu ARX má tvar

$$y(k) = -a_1 y(k-1) - \dots - a_n y(k-n) + b_0(k) + \dots b_m(k-m) + v(k) \quad (27)$$

b) **OE** (Output Error – pokud budeme volit $GF(z^{-1}) = A(z^{-1})$)



Obr.3.4 Model OE

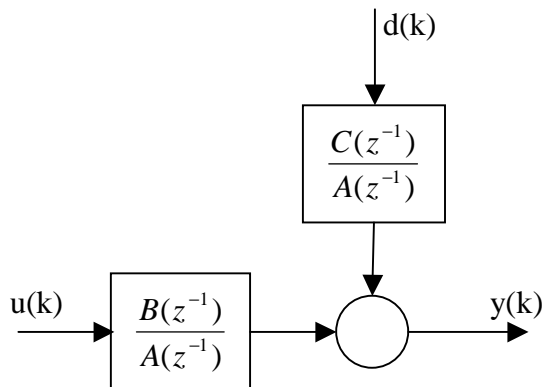
Výstup OE modelu vypadá takto

$$y(k) = \frac{B(z^{-1})}{A(z^{-1})} u(k) + v(k) \quad (28)$$

a jeho diferenční rovnice

$$y(k) = -a_1 y(k-1) - \dots - a_n y(k-n) + b_0(k) + \dots + b_m(k-m) + v(k) + a_1 v(k-1) + \dots + a_n v(k-n) \quad (29)$$

c) **ARMAX** (Auto-Regressive Moving Average with eXogenous variable), pokud volíme $GF(z^{-1}) = C(z^{-1})$. Výstup modelu je



Obr.3.5 Model ARMAX

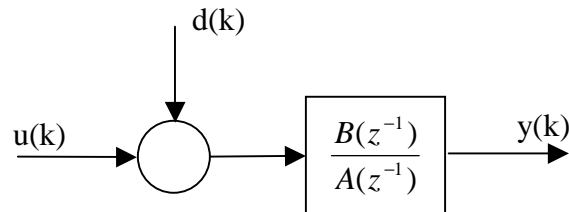
$$y(k) = \frac{B(z^{-1})}{A(z^{-1})} u(k) + \frac{C(z^{-1})}{A(z^{-1})} d(k) \quad (30)$$

jeho diferenční rovnice má tvar

$$y(k) = -a_1 y(k-1) - \dots - a_n y(k-n) + b_0(k) + \dots + b_m(k-m) + c_0 v(k) + c_1 v(k-1) + \dots + c_n v(k-n) \quad (31)$$

d) **IN** (Input error) – pokud $GF(z^{-1}) = B(z^{-1}) / A(z^{-1})$. Porucha se přičítá k akční veličině

$$y(k) = \frac{B(z^{-1})}{A(z^{-1})} [u(k) + v(k)] \quad (32)$$



Obr.3.6 Model IN

4. Estimace

Pokud chceme řídit reálný systém, tak v podstatě nikdy není možné změřit všechny vnitřní stavy. V diplomové práci je diskutována regulace SISO systému. SISO je systém, který má jednu vstupní(akční) veličinu $u(k)$ a jednu výstupní veličinu $y(k)$. Na systému je tedy možné měřit pouze výstupní veličinu.

Výpočet vektoru stavu za uvedených podmínek nazýváme estimace a prvek – algoritmus, který tento odhad realizuje je nazýván estimátor. Estimátor odhaduje stav systému v případě vstupu determinovaných i náhodných poruch, vstupujících do systému, nebo na jeho výstupní veličiny.

Jsou používány dva typy estimátorů: a) Kalmanův estimátor

c) Luenbergerův estimátor

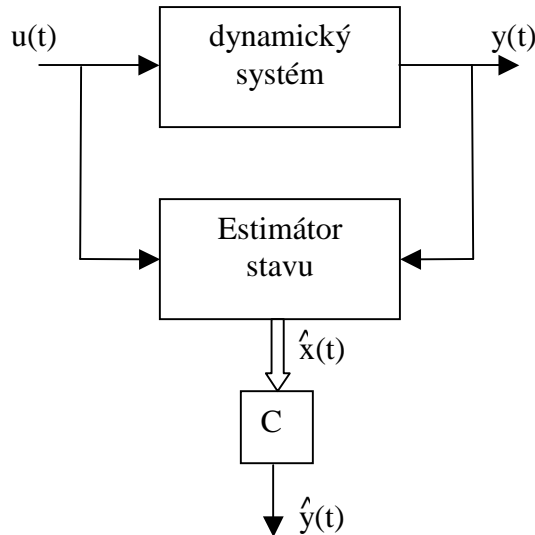
- a) Tento typ estimátoru se používá jestliže měřený signál $y(k)$ obsahuje aditivní šum. Kalmanův estimátor potom nastavujeme minimalizací kritéria

$$J = \sum_{k=-\infty}^{\infty} \left(x(k) - \hat{x}(k) \right)^2 \quad (33)$$

,kde $\hat{x}(k)$ je odhadovaný stavový vektor

$x(k)$ je stavový vektor regulované soustavy

- b) Pokud na měřenou poruch nepůsobí šumový signál, je možné předpokládat, že nepůsobí ani na stavové veličiny a je tedy možno použít tzv. **deterministický estimátor**



Obr.4.1 Blokové zapojení estimátoru a systému

,kde \hat{y} a \hat{x} jsou odhadované (estimované) veličiny

Dále již bude diskutován pouze diskrétní deterministický estimátor stavu.

Deterministický estimátor stavu

Schéma zapojení estimátoru je uvedeno na obrázku. Popisovaný estimátor byl navržen D. G. Luenbergerem a vychází z předpokladu, že je znám lineární stavový model. Dále je nutné, aby byl měřen výstup ze soustavy. Stavové rovnice jsou ve tvaru:

$$\begin{aligned} x(k+1) &= Mx(k) + Nu(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned} \quad (34)$$

Jako základu pro konstrukci estimátoru se využívá přímo tvaru modelu systému.

Estimátor pak popíšeme rovnicí

$$\begin{aligned} \hat{x}(k+1) &= M_E \hat{x}(k) + N_E u(k) \\ \hat{y}(k) &= C \hat{x}(k) \end{aligned} \quad (35)$$

,kde M_E a N_E jsou matice estimátoru

$\hat{x}(k)$ a $\hat{y}(k)$ jsou odhadovaný výstup a stav

Pokud zavedeme chybu estimace

$$\begin{aligned} \Delta x(k+1) &= x(k+1) - \hat{x}(k+1) = Mx(k) + Nu(k) - M_E \hat{x}(k) - N_E u(k) = \\ &= Mx(k) - M_E \hat{x}(k) + (N - N_E)u(k) \end{aligned} \quad (36)$$

Dále aby chyba a tedy ani odhad nezávisely na akční veličině $u(k)$, musí být splněna rovnost $N = N_E$. Podobně nechceme, aby byla chyba estimace závislá na vektoru $x(k)$ a jeho odhadu $\hat{x}(k)$, proto pokládáme $M = M_E$. Chyba estimace je pak rovna výrazu

$$\Delta x(k+1) = M \cdot \Delta x(k) \quad (37)$$

Je jasné, že jestliže bude $\Delta x \neq 0$, pak je estimace podle (37) dynamický proces, který závisí na matici systému M . Takovýto estimátor však nevyhovuje požadavkům. Zavádíme proto korekci estimace. Informace o stavu estimace potom vypadá následovně

$$\Delta y(k) = y(k) - \hat{y}(k) \quad (38)$$

Tento rozdíl využíváme a zavádíme ho do estimátoru přes korekční matici L . Rovnice estimátoru má pak tvar

$$\hat{x}(k+1) = M_E \hat{x}(k) + N_E u(k) + L(y(k) - \hat{y}(k)) \quad (39)$$

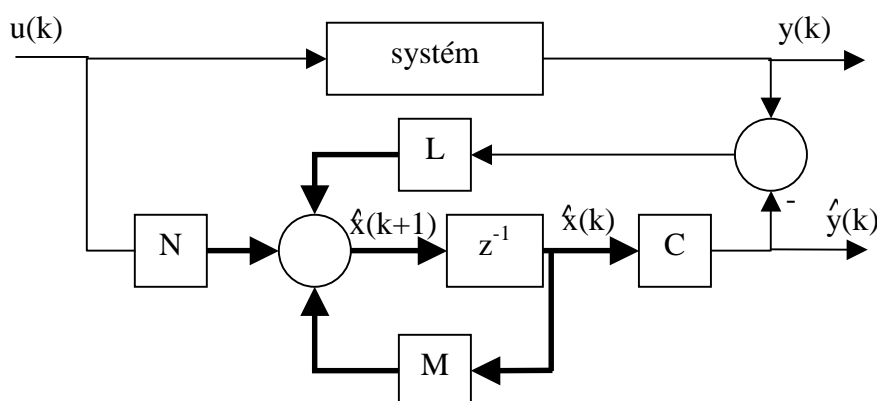
Chyba odhadu je poté rovna

$$\Delta x(k+1) = x(k+1) - \hat{x}(k+1) = Mx(k) + Nu(k) - \{M_E x(k) + N_E u(k) + L(y(k) - C\hat{x}(k))\} \quad (40)$$

Po úpravách a zavedení rovností, které jsme předpokládali již u předchozích rovnic, a to $N_E = N$ a $M_E = M$, dostaneme výslednou chybu odhadu

$$\Delta x(k+1) = (M - LC)\Delta x(k) \quad (41)$$

,kde L je prozatím neurčená matice, která jak je vidět má vliv na dynamiku estimátoru. Tuto matici volíme tak, aby výsledná chyba estimace konvergovala k nule.



Obr.4.2 Struktura estimátoru s korekční maticí

Dynamické vlastnosti chyby estimace jsou dány vlastními čísly matice $(M - LC)$, tyto můžeme vypočítat jako determinant výrazu $\{zI - (M - LC)\}$.

$$\det\{zI - (M - LC)\} = z^n + a_{n-1}z^{n-1} + \dots + a_1z + a_0 \quad (42)$$

,kde I je jednotková matice

z je komplexní proměnná v z -transformaci

Tento determinant se nám dále zjednoduší, pokud bude mít matice M speciální tvar, například normální formu rekonstruovatelnosti.

Matici koeficienty matice L potom vypočteme tak, že koeficienty u mocnin proměnné z , které můžeme ovlivnit maticí L , položíme rovny nule. Výsledkem budou vlastní čísla matice $(M - LC)$. S takto vypočítanou maticí L bude chyba estimace nejrychleji konvergovat k nule.

Estimátor pro bezchybný odhad vektoru stavu

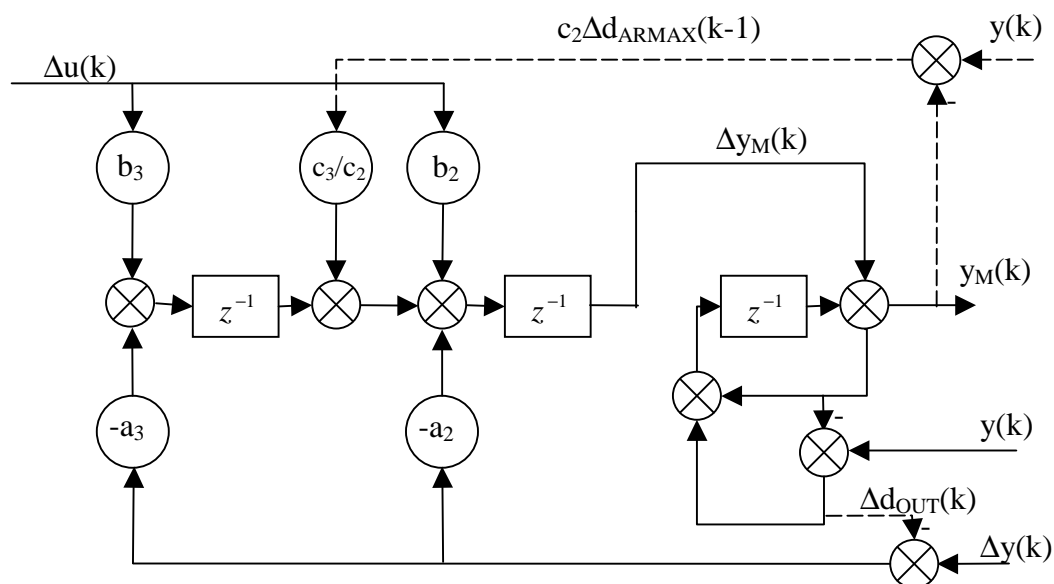
Tento estimátor byl navržen na TU v Liberci panem Prof. B. Hanušem. Jeho návrh vychází z klasického luenbegerova estimátoru stavu, který byl modifikován. Upravený estimátor je schopen odhadovat stav soustavy úplně bezchybně, v případě, že do systému vstoupí porucha, na kterou je estimátor upraven. Chyba estimace v tomto případě je rovna pouze impulsu, který vzniká v prvním časovém okamžiku od vstupu poruchy do soustavy. Tato chyba je způsobena tím, že vlastně až v druhé kroku dostává estimátor informaci o výstupu soustavy, který byl pozměněn vstoupivší poruchou. Čili velikost tohoto impulsu je dána rozdílem

$$\hat{y} - y = y \cdot z^{-1} + \Delta \hat{y} - (y \cdot z^{-1} + \Delta y) = \Delta \hat{y} - \Delta y \quad (43)$$

,kde $y(k)$ je výstup soustavy, který je již ovlivněn poruchou

 $\hat{y}(k)$ je výstup estimátoru

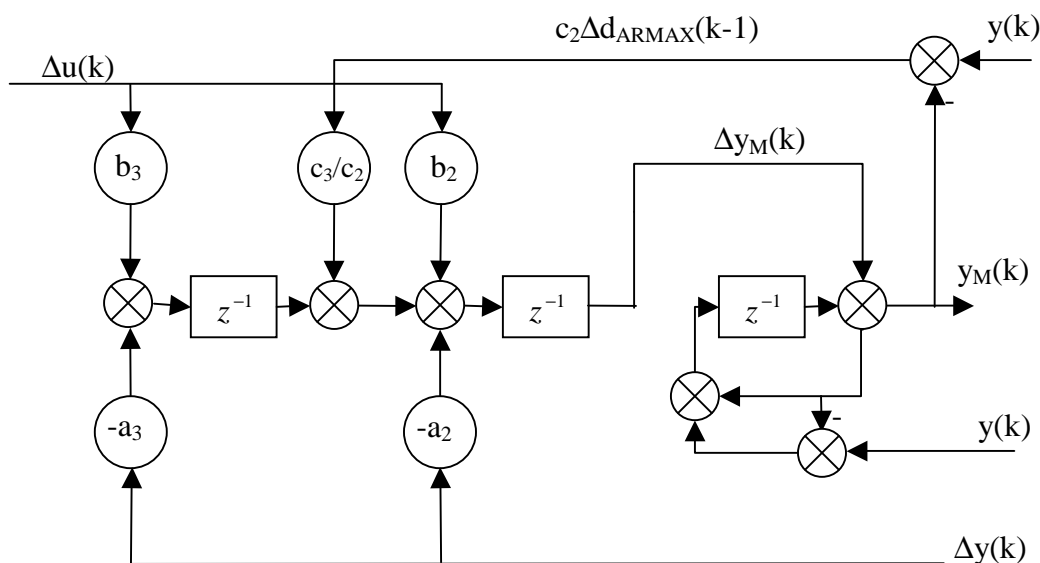
V případě vstupu jiné poruchy, než na kterou je estimátor naladěný, vznikne chyba estimace, která bude konvergovat k nule a bude odstraněna v konečném počtu kroků.



Obr.4.3 Diskutovaný estimátor

Vraťme se ještě k Obr.2.5, kde je integrátor zapojen na vstup soustavy. V řešeném příkladě je ale integrátor v estimátoru stavu zapojen na výstup modelu soustavy. Tato struktura estimátoru umožňuje jednoduše zadávat žádanou hodnotu regulované – výstupní veličiny systému.

Pro bezchybnou estimaci poruch IN, ARX a ARMAX je obecné schéma z Obr.4.3 upraveno, jak je vidět na Obr.4.4.



Obr.4.4 Estimátor pro in, arx, armax

Korekční matice L je spočtena jako podíl b_n / b_2 pro estimátor IN, nulová pro estimátor ARX a c_n / c_2 pro estimátor ARMAX. Chybu estimace v tomto zapojení můžeme vyjádřit z (43) jako

$$\begin{aligned}
 y_{E1} - y &= \Delta y_{E1} - \Delta y = B_E \Delta u + (1 - A_E) \Delta y - \Delta y = \\
 &= B_E \Delta u - A_E \left(\frac{B}{A} \Delta u + \frac{C}{A} \Delta d_{ARMAX} + \frac{1}{C} \Delta d_{ARX} + \Delta d_{OUT} \right) = \\
 &= \left(B_E - \frac{A_E}{A} B \right) \Delta u - \frac{A_E}{A} C \Delta d_{ARMAX} - \frac{A_E}{C} \Delta d_{ARX} - A_E \Delta d_{OUT}
 \end{aligned} \tag{44}$$

,kde y_{E1} je výstupní veličina estimátoru naladěného na poruchu IN, ARX nebo ARMAX

A_E a B_E jsou polynomy estimátoru

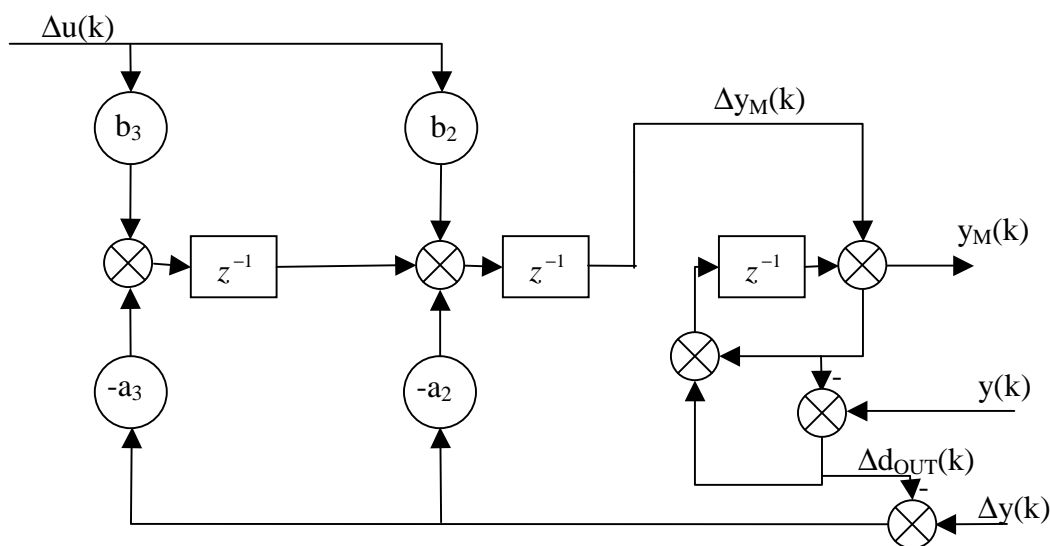
A , B jsou polynomy soustavy

C je polynom modelu poruchy

Za předpokladu, že vstupuje porucha Δd_{ARMAX} do estimátoru naladěného na poruchu ARMAX, můžeme postup estimace popsat asi takto.

Porucha vstupuje do soustavy v k -tém kroku násobená polynomem C . V $k+1$ kroku se dostává na výstup soustavy, prochází přes diskretní integrátor a dostává se přes zpětnou vazbu na vstup estimátoru. Hodnota signálu na této zpětné vazbě v $k+1$ kroku je rovna výrazu $c_2 \cdot \Delta d_{ARMAX}$. Pokud chceme, aby estimátor odhadoval stav bezchybně, musí na něm být stejný stav jako na soustavě, proto zavádíme zmiňovanou zpětnou vazbu před zpožďovací člen a signál dělíme členem c_2 a zároveň násobíme členem c_3 . Tím pádem dostaneme na estimátoru stejný stav jako na soustavě. Na výstupu estimátoru je stále y_M , které odpovídá výstupní veličině soustavy ještě před vstupem poruchy a tento rozdíl $y_M - y$ tvoří impulsovou chybu estimace.

Pro poruchy IN a ARX je postup estimace stejný. Je třeba pouze dosadit příslušnou korekční matici L .



Obr.4.5 estimátor pro out

Pro estimace poruchy OUT je schéma estimátoru z Obr.4.3. upraveno podle Obr.4.5. Chybu estimace můžeme opět vyjádřit z rovnice (43).

$$\begin{aligned}
 y_{OUT} - y &= \Delta y_{OUT} - \Delta y = B_E \Delta u + (1 - A_E)(\Delta y - (y - y_{OUT})) - \Delta y \\
 A_E (y_{OUT} - y) &= B_E \Delta u + (1 - A_E)\Delta y - \Delta y = B_E \Delta u - A_E \left(\frac{B}{A} \Delta u + \frac{C}{A} \Delta d_{ARMAX} + \Delta d_{OUT} \right) \\
 y_{OUT} - y &= \left(\frac{B_E}{A_E} - \frac{B}{A} \right) \Delta u - \frac{C}{A} \Delta d_{ARMAX} - \Delta d_{OUT}
 \end{aligned} \tag{45}$$

,kde y_{OUT} je výstupní veličina estimátoru naladěného na poruchu OUT

A_E a B_E jsou polynomy estimátoru

A , B jsou polynomy soustavy

C je polynom modelu poruchy

Postup estimace poruchy OUT pomocí estimátoru *out* vypadá následovně.

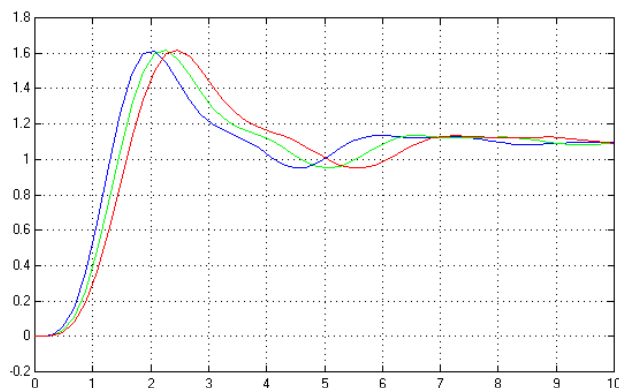
Porucha nevstupuje do soustavy, ale až na její výstup, takže vnitřní stavové veličiny vůbec neovlivní. Přičítá se tedy k výstupu ze soustavy a prochází přes diskretní derivační člen. Zároveň se dostává do soustavy sumátorů u estimátoru, kde se vlastně tato porucha zpět odečte a do zpětné vazby estimátoru vstupuje pouze Δy . Proto zůstávají vnitřní stavové veličiny na estimátoru a na soustavě shodné.

Další zvláštností toho zapojení je, že informaci o první a druhé stavové veličině získává regulátor přímo z výstupu regulované soustavy. V důsledku toho mají tyto veličiny vždy správnou hodnotu.

Časová afinita

Pro účely simulace zavádíme pojem časová afinita. Jedná se o lineární transformaci soustavy, kterou provedeme tak, že póly a nuly (kořeny jmenovatele a čitatele) soustavy dělíme koeficientem. Tím získáme soustavu s podobnou dynamikou, která se bude lišit od nominální soustavy v závislosti na velikosti koeficientu, kterým jsme dělili. Nová soustava je

„rychlejší“, pokud jsme koeficient transformace zvolili menší než jedna nebo „pomalejší“ pokud jsme koeficient zvolili větší než jedna. Dále se takto transformovaná soustava liší v zesílení, pokud však chceme využít pouze časovou afinitu je toto nežádoucí. Proto získaný přenos vydělíme ještě jeho vlastním zesílením(viz. Obr.4.6).



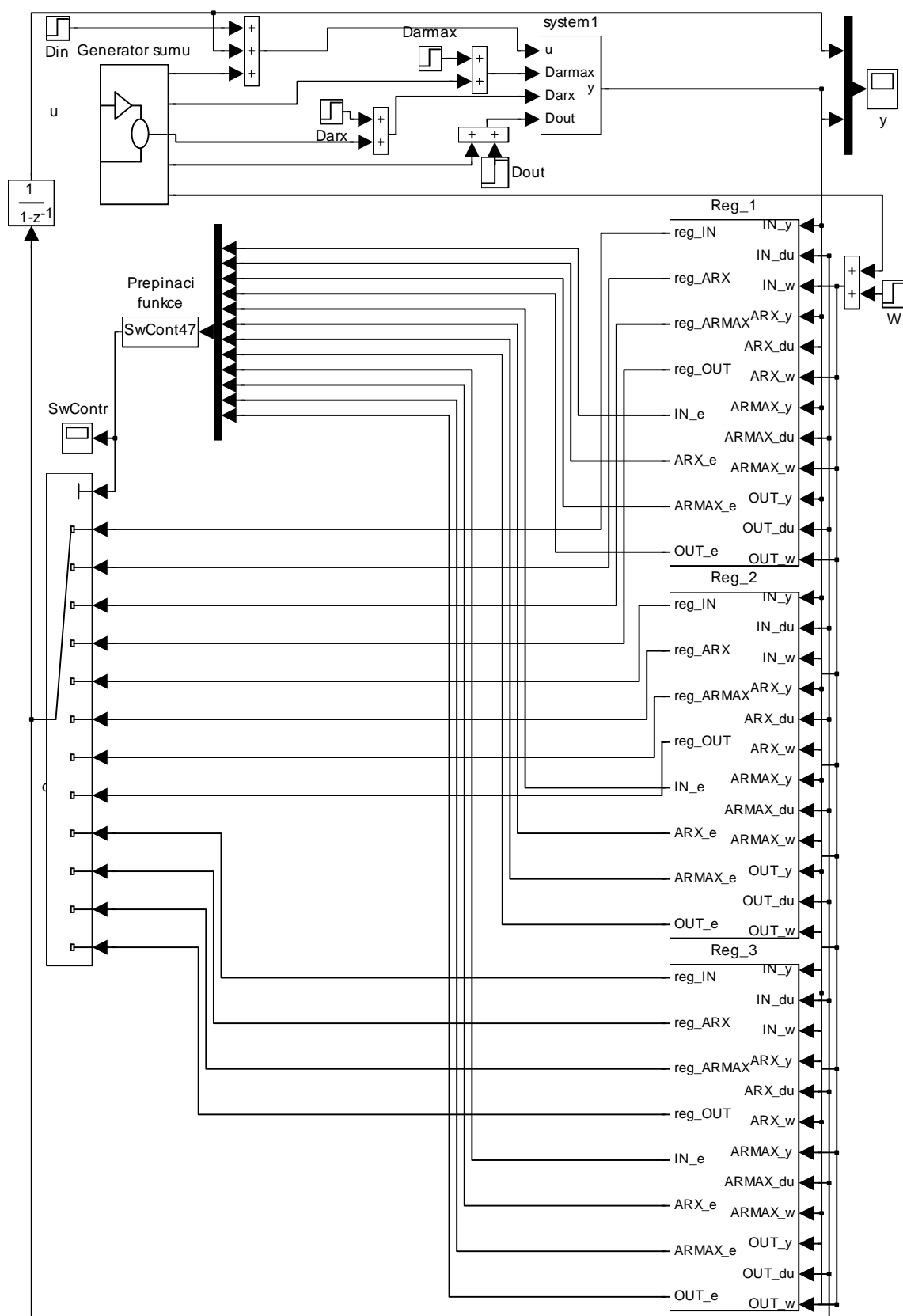
Obr.4.6 Časově afinní soustavy

5. Regulátor s proměnnou strukturou

Pro zlepšení regulace a rozšíření oblasti robustnosti, byl sestaven stavový regulátor s proměnnou strukturou. Systém se sestává z několika různě naladěných regulátorů a již zmiňovaných estimátorů, pro bezchybnou estimaci, a přepínacího bloku. Jeho struktura je uspořádána tak, aby se dalo přepínat mezi jednotlivými bloky regulátorů s estimátory podle námi stanovených pravidel.

V našem případě byly pro výpočet regulátorů zvoleny tři koeficienty časové afinity. Dále je ve struktuře použito čtyř různých typů estimátorů (IN, ARX, ARMAX, OUT). Celkem je tedy použito dvanáct regulátorů a dvanáct estimátorů, které jsou uspořádány do skupin tak, že v každé skupině jsou čtyři různé estimátory se čtyřmi regulátory, které jsou však nastaveny na stejný koeficient časové afinity.

Hodnota koeficientů byla zvolena 0,9 – 1 – 1,1



Obr.5.1. Schéma zapojení ze Simulinku

, kde Din	porucha IN
Darx	porucha ARX
Darmax	porucha ARMAX
Dout	porucha OUT
přepínací funkce	přepíná pomocí bloku „switch“ mezi jednotlivými regulátory s estimátory
generátor šumu	slouží k testování přepínací funkce
system1	diskrétní systém, implementován do schématu jako subsystém, z důvodů úspory místa
y (scope).....	osciloskop, ke sledování průběhu regulace (průběh y a u)
w	změna žádané hodnoty
Δu	přírůstek akční veličiny
$\frac{1}{1-z^{-1}}$	diskrétní integrátor
Reg1, Reg2, Reg3	bloky čtyř regulátorů a estimátorů, které jsou nastaveny na stejný koeficient časové afinity

6. Simulační program – Matlab

Pro vytvoření modelu soustavy a jeho řízení bylo využito softwarového produktu firmy MathWorks, programu MatLab verze 5.3. Tento program je velmi komplexním nástrojem pro řešení problémů z celé řady přírodovědných oborů. Skládá se ze základního jádra, které tvoří Matlab-Simulink, ke kterému je možno po zakoupení doinstalovat libovolný počet tzv. toolboxů s různým zaměřením.

Základní těleso MATLABu obsahuje funkce orientované pro práci s maticemi a bohatá knihovna Toolboxů umožňuje řešení široké škály problémů, jako jsou: teorie řízení, optimalizace, zpracování signálů, identifikace a mnohé další. Srdcem programu jsou algoritmy pro operace s maticemi komplexních čísel, nejen běžné operace jako násobení, inverze, determinant, ale i jako maticový kalkulátor, protože všechny matice se zapisují do operační paměti (workspace). Součástí MATLABu je široká škála funkcí pro výpočty, ukládání, statistické zpracování, vizualizaci dat, značný počet dostupných, problémově orientovaných balíčků již hotových funkcí, jednoduchá syntaxe a kvalitní implementované algoritmy.

Funkce je obvykle složitější činnost, která jeden nebo více vstupních parametrů zpracuje do jednoho nebo více výstupních parametrů podle určitého předpisu (algoritmu). Pokud potřebujeme určitou posloupnost příkazů opakovat, tento zápis se uloží na disk do souboru s příponou **m**, nazýváme ho m-file.

Funkce v Matlabu

K přednostem MATLABu patří možnost rozšiřování o vlastní funkce, což při programování výrazně urychluje práci a zároveň je takto psaný program lépe čitelný. První řádek funkce musí obsahovat hlavičku, která má formát:

function [výstup1,výstup2,...] = *jméno funkce* (vstup1,vstup2,...)

a zajišťuje přenos dat do a z funkce. Dále je nutné, pokud chceme funkci uložit do externího souboru, aby bylo jméno tohoto souboru stejné jako jméno funkce. Na dalších řádcích se píší příkazy funkce (tzv. tělo funkce). Pokud používáme nějaké globální proměnné, je nutné jejich deklaraci uvádět ve všech funkcích znovu.

Simulink

Simulink je prostředek pro simulaci a modelování dynamických systémů, který využívá algoritmy MATLABu pro numerické řešení diferenciálních rovnic. Umožňuje snadné a rychlé vytváření modelů, prakticky jakýchkoliv, soustav pomocí spojování bloků s různou funkcí.

Knihovna simulinku obsahuje množství předdefinovaných bloků popisujících lineární, nelineární, spojité a diskrétní systémy a umožňuje použít širokou škálu vstupních signálů. Výstupní data z modelovaného systému mohou být vizualizována přímo, pomocí grafických bloků, nebo mohou být uložena do datového souboru, který má příponu **mat**.. Simulink umožňuje rozšiřovat knihovnu o vlastní funkční bloky. Každá standardní knihovna obsahuje jednotlivé bloky, které lze myší přetáhnout do okna kde vytváříme své schéma. Jednotlivé bloky uspořádány do skupin podle oblasti své funkce..

Standardní knihovny

Sources - Obsahuje bloky použitelné jako zdroje běžných druhů testovacích signálů jako je jednotkový skok, rampa, sinus atd.

Sinks – Tato knihovna obsahuje nástroje pro zobrazení průběhů signálů. Asi nejčastěji používaným je Scope – osciloskop.

Continuous - Zahrnuje bloky sloužící pro vytvoření spojitých dynamických modelů, zároveň také pro operace se spojitým signálem.

Discrete – Obsahuje podobné funkce jako předchozí knihovna, ale pro diskrétní popis.

Math - Bloky z této knihovny jsou nezbytné pro matematické a logické operace. Nejvíce používaným blokem je Sum pro součet. Pro nás byl velmi užitečný blok Matrix Gain, který umožňuje maticové násobení, pokud jsou signály tvořeny vektory, použili jsme ho k popisu dynamického systému ve stavovém prostoru a pro návrh regulátoru.

Functions & Tables - Knihovna obsahující bloky pro funkce a tabulky. Byl použit blok S-function do nějž byl naprogramován algoritmus přepínání.

Nonlinear - Slouží pro konstrukci a simulaci nelineárních soustav. Použili jsme blok Multiport Switc, který umožňuje přepínání mezi libovolným počtem signálů..

Signals & Systems - Tato knihovna obsahuje bloky pro slučování(Mux) a rozdělování(Demux) signálů, ukládání a čtení proměnných a další pomocné bloky.

S-funkce

M-file s S-funkcí slouží k vytváření vlastních diferenciálních rovnic, diskrétních systémů rovnic, nebo algoritmů, které mohou být použity s blokovým schématem ze Simulinku.

Velkou výhodou této funkce je, že může být volána prostřednictvím simulačního schématu v každém kroku simulace a může jí být předán libovolně dlouhý vektor hodnot pro zpracování.

Obecný tvar S-funkce

$[SYS, X0, STR, TS] = SFUNC(T, X, U, FLAG, P1 \dots Pn)$

To co S-funkce vrací v daný časový okamžik T, závisí na hodnotě tzv. Flagu (návěští), stavovém vektoru X a vstupním vektoru Y.

FLAG	POPIS
0	Iniciace, vrací velikost systému v SYS, počáteční stav v X0,
1	Vrací derivace spojitých stavových veličin v SYS
2	Obnovuje diskrétní stavy $SYS = X(n+1)$
3	Vrací výstup v SYS

4	Vrací následující časový okamžik pro variabilní vzorkovací krok v SYS
5	Rezervováno
6	Přerušení, vymazání SYS = []

Stavové vektory, X a X0 se skládají ze spojitých stavů, které následují diskrétní stavy. Volitelné parametry, P1 ... Pn deklarovány v S-funkci a použity během jakéhokoliv Flagu.

Když je S-funkce zavolána s FLAG = 0, vrátí následující informace.

SYS(1) = počet spojitých stavových veličin

SYS(2) = počet diskrétních stavových veličin

SYS(3) = počet výstupů

SYS(4) = počet vstupů

Jakýkoliv z prvních čtyřech elementů v SYS mohou být rovny -1, což znamená, že jsou deklarovány dynamicky. Aktuální délka pro ostatní flagy bude rovna délce vstupu U.

SYS(5) = rezervováno, musí být 0

SYS(6) = zapínání přímého plnění (1=ano 0=ne). Přímé plnění jestliže je U použito během volání Flag = 3. Nastavením na nulu deklarujeme, že nebudeme používat U během Flag = 3. Pokud je toto porušeno nastane nepředvídatelný stav.

SYS(7) = počet časových vzorků, což je ekvivalentní počtu řádků v matici TS.

X0 = počáteční stavové podmínky nebo [] (prázdná matice) pokud nejsou žádné

STR = řetězec pořadí jednotlivých stavů, který je specifikován jako []

TS = m x 2 matice obsahující informaci o vzorkovacím čase (perioda a offset), kde m je počet vzorkovacích časů.

Pro naši simulaci jsem však využil pouze některé Flagy (návěští) a to :

0 – pro nastavení počátečních podmínek a inicializaci proměnných

2 – ve kterém sice nic není, ale funkce bez něho nefungovala

3 – hlavní rozhodovací algoritmus

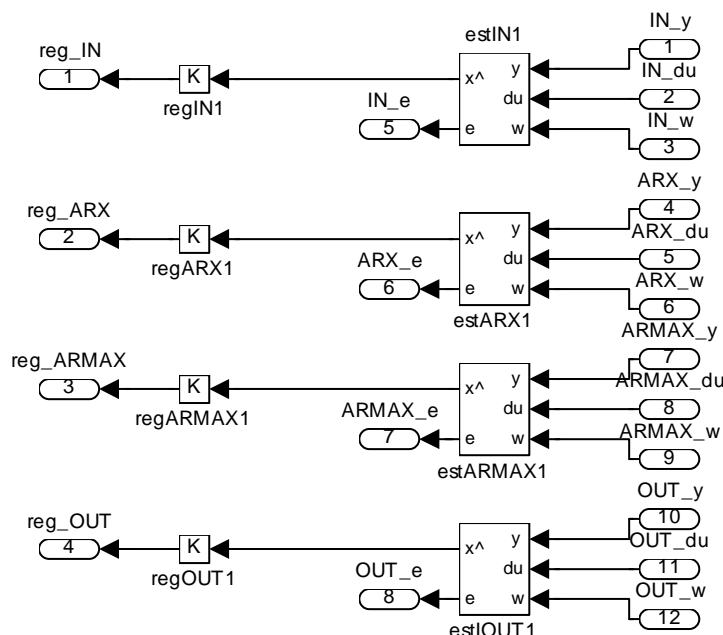
9 – smazání SYS

7. Realizace simulačního modelu

Jak již bylo zmíněno, simulační model byl vytvořen pomocí blokového schématu v Simuliku. Jeho struktura je na Obr.5.1 . Pro přehlednost bylo použito několika subsystémů.

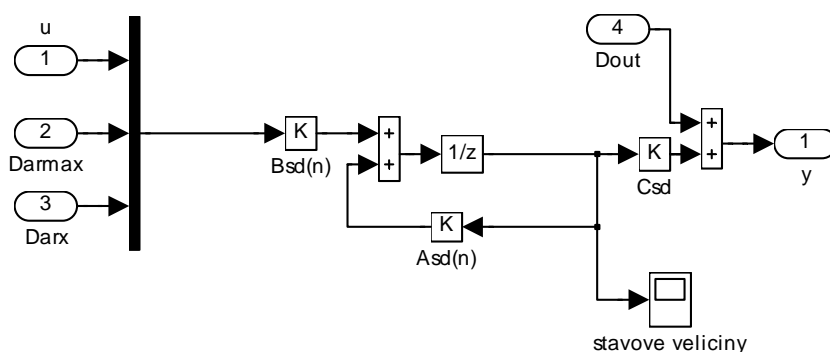
Subsystémy Reg1 – 3 obsahují vždy čtyři regulátory, které jsou stejné a k nim připojené estimátory, z nich každý je naladěn na jednu z poruch : IN, ARX, ARMAX, OUT viz.

Obr.7.1. Přičemž všechny členy v daném bloku jsou naladěny na soustavu se stejným koeficientem afinity (0,9 - 1 - 1,1).



Obr.7.1 Blok Regx

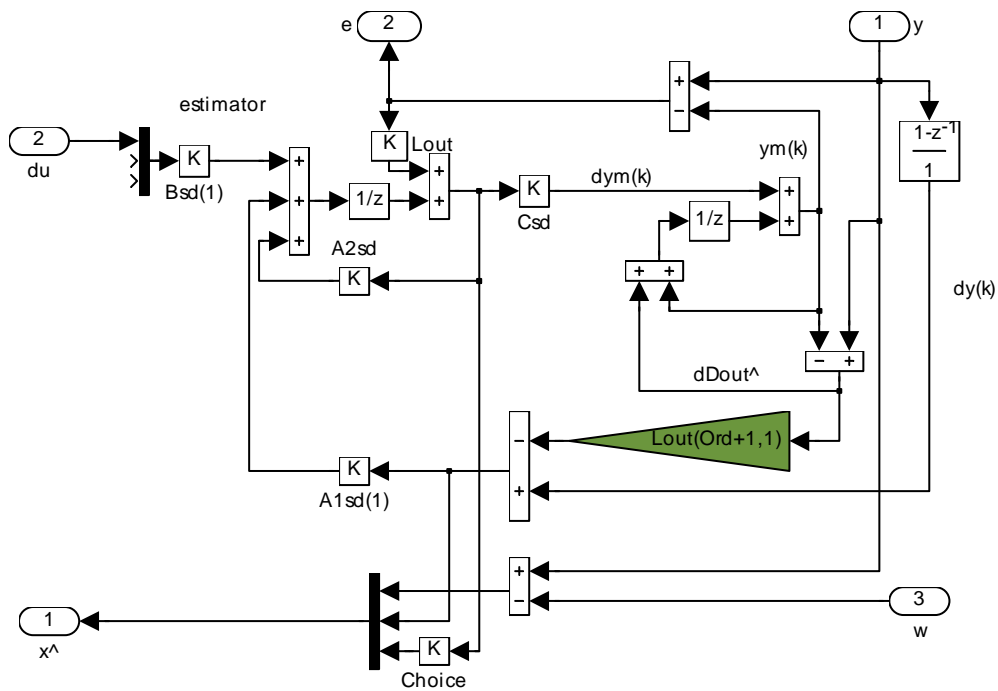
Dalším použitým subsystémem je blok, ve kterém je model regulované soustavy, viz Obr.7.2. Jedná se o klasický stavový popis, který je nepatrně upraven. Vstup modelu je přizpůsoben tak, aby bylo možno simulovat poruchy IN(což je ekvivalentní vstup s Δu), ARX a ARMAX. Matice B má proto dva sloupce navíc.



Obr.7.2 Stavový popis soustavy v Simulinku

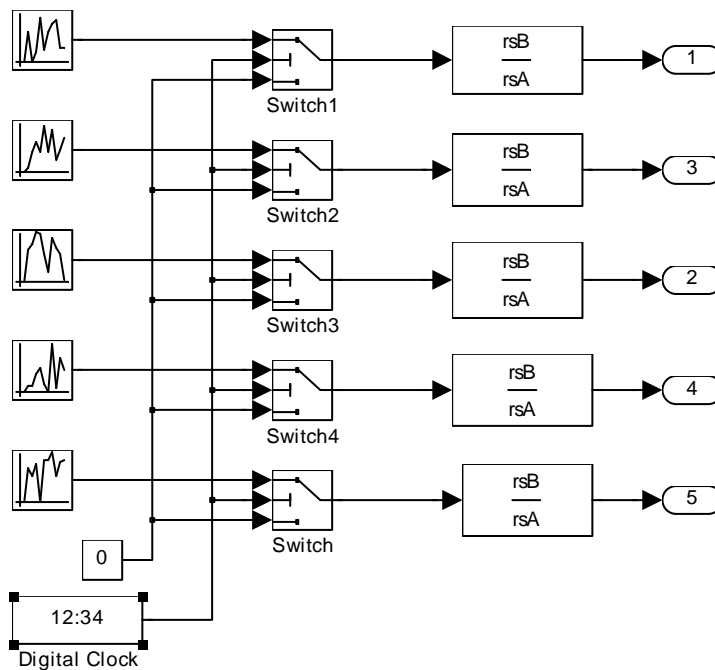
Jednotlivé estimátory jsou také uzavřeny v subsystémech. Jsou zkonstruovány tak, že mají všechny stejnou strukturu. Do všech estimátorů byl umístěn blok, který spojí nebo rozpojí zpětnou vazbu, podle typu estimátoru. Tento „spínač“ je možné řídit posledním koeficientem matice L. Do tohoto korekčního vektoru byl přidán jeden koeficient, který je roven nule v případě estimátorů IN, ARX, ARMAX a v případě estimátoru OUT je nastaven na jedničku. Na následujícím obrázku Obr.7.3. je tento „spínač“ zvýrazněn zeleně. Dále je zde matice Choice, která zajišťuje výběr 3-tího až n-tého stavu, kde $n = \text{řád soustavy} + 1$, protože jak již bylo zmíněno, první a druhý stav je do regulátoru zaveden přímo z regulované soustavy.

U mutliplexoru před maticí B nejsou zapojeny všechny vstupy, protože je použita matice B ze stavového popisu soustavy a tam tyto vstupy slouží pro vstup poruchy ARX a ARMAX. U estimátoru je pro funkčnost programu nutné vstupy zachovat i když nejsou zapojeny.



Obr.7.3 Struktura estimátoru

Dále byl pro účely simulace vytvořen generátor náhodných signálů viz. Obr.7.4 , který může generovat až pět signálů nezávisle na sobě najednou. Základem je generátor náhodných čísel, který v každý vzorkovací interval vygeneruje jedno číslo. Tento signál je poté filtrován přes filtr druhého řádu, který zajistí rozumný průběh signálu. Pomocí hodin a přepínače je možno takto vytvořený signál kdykoliv spustit.



Obr.7.4 generátor náhodného signálu.

Princip přepínání mezi regulátory

Algoritmus přepínací funkce je založen na analýze chyb jednotlivých estimátorů, které jsou vyjádřeny rovnicí.

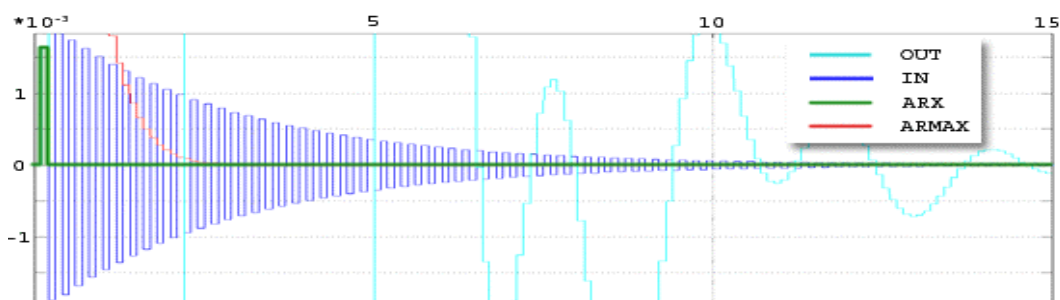
$$e_i = y_M - y \quad (46)$$

, kde e_i je chyba i-tého estimátoru

y_M je estimovaný stav soustavy

y je skutečný stav výstupu soustavy.

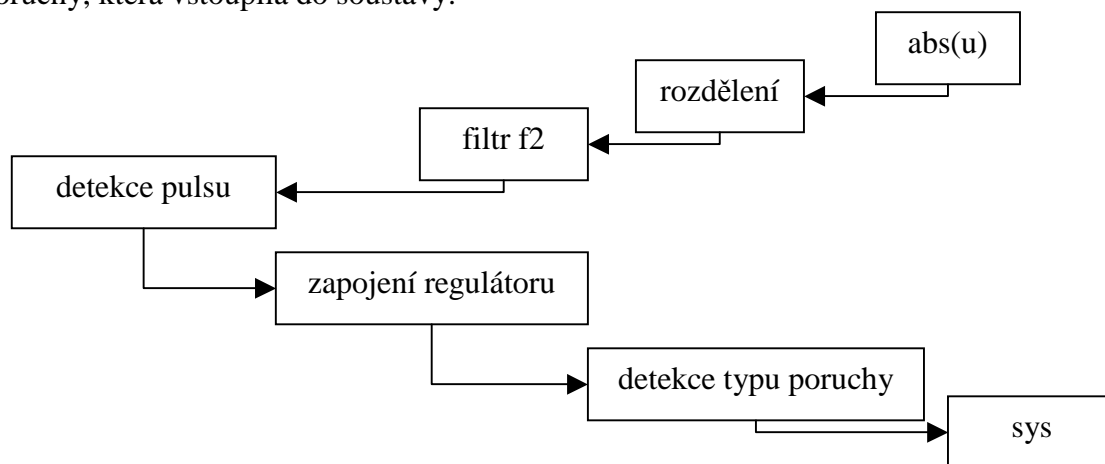
Po zpracování a vyhodnocení všech chyb jednotlivých estimátorů se na výstupu přepínací funkce objeví hodnota od jedné do dvanácti. Podle této hodnoty je zapojen, pomocí bloku *switch*, odpovídající regulátor do zpětné vazby.



Obr.7.5 Chyba estimace estimátoru arx při vstupu poruchy arx

Algoritmus přepínací funkce

Navržený algoritmus je naprogramován v tzv. S-funkci. a pracuje s absolutními hodnotami chyb estimace. Využívá toho, že pokud vstoupí do soustavy porucha, tak chyba estimátoru, který je na tuto poruchu naladěný je pouze impuls, viz. Obr.7.5. V případě, že se polynomy estimátoru a soustavy poněkud liší, objeví se v grafu chyby tohoto estimátoru těž impuls. Další hodnoty chyby estimace jsou potom již podstatně menší. Tento jev se však projevuje pouze u estimátorů naladěných na daný typ poruchy. Proto je možné ho využít k identifikaci poruchy, která vstoupila do soustavy.



Obr.7.6 schéma přepínacího algoritmu

Blok ABS(u) – jak již bylo řečeno algoritmus přepínání pracuje s absolutními hodnotami.

V tomto úseku program jsou chyby estimace převedeny na absolutní hodnoty.

$edu = abs(u);$

Blok rozdělení – pomocí cyklu jsou absolutní hodnoty rozděleny podle typu estimátoru

```
for I=1:3;
    ein(I)=edu((I-1)*4+1);      earx(I) =edu((I-1)*4+2);
    earmax(I)=edu((I-1)*4+3);   eout(I) =edu( I*4    );
end;
```

Blok filtr F2 – je využíván až dále v programu, ale jeho hodnoty se plní již zde. Koeficient, kterým se násobí hodnota filtru je volen přibližně tak, aby křivka, která vznikne, byla jakousi obalovou křivkou pro absolutní hodnoty chyb jednotlivých estimátorů.

```
for I=1:12
    if (eduf2(I)*0.95>edu(I) & eduf2(I)>0)
        eduf2(I)=eduf2(I)*0.95;
    else
        eduf2(I)=edu(I);
    end;
end;
```

Blok detekce poruchy – slouží pro detekci typu poruchy. Využívá již zmiňovaného impulsu chyby estimace, který vznikne, když vstoupí porucha, na kterou je estimátor naladěn. V prvním cyklu algoritmus hledá nárůst chyby estimace a zaznamená výsledky do matice START. V druhé cyklu je poté rozhodnuto o kterou poruchu se jedná.

```
rozptyl=0.15;
for I=1:4
    if (STARTOLD(I,1)+STARTOLD(I,2)+STARTOLD(I,3))==3
        STARTOLD(I,:)=[0.35 0.35 0.35];
    else
        STARTOLD(I,:)=[0 0 0];
    end
end
for I=1:4
    if porucha~=I
        h1 = (edu(I,1)-edu(I,2));
        h2 = (edu(I+4,1)-edu(I+4,2));
        h3 = (edu(I+8,1)-edu(I+8,2));
        stred=mean([h1 h2 h3]);
        if stred>0
            if (h1>(1-rozptyl)*stred & h1<(1+rozptyl)*stred)
                STARTOLD(I,1)=1;
            end;
            if (h2>(1-rozptyl)*stred & h2<(1+rozptyl)*stred)
                STARTOLD(I,2)=1;
            end;
            if (h3>(1-rozptyl)*stred & h3<(1+rozptyl)*stred)
                STARTOLD(I,3)=1;
            end;
        end;
    end;
end;
```

```

end;
hh1 = edu(I,2)-edu(I,1);
hh2 = edu(I+4,2)-edu(I+4,1);
hh3 = edu(I+8,2)-edu(I+8,1);
hstred = mean([hh1 hh2 hh3]);
hrozptyl=0.2;
pokles=0.35;
if hh1>0 & hh2>0 & hh3>0
if (hh1>(1-hrozptyl)*hstred & hh1<(1+hrozptyl)*hstred)
if (edu(I,2)*pokles>edu(I,1) & edu(I,2)>0)
START(I,1)=START(I,1)+1;
end;
end;
end;

if (hh2>(1-hrozptyl)*hstred & hh2<(1+hrozptyl)*hstred)
if (edu(I+4,2)*pokles>edu(I+4,1) & edu(I+4,2)>0)
START(I,2)=START(I,2)+1;
end;
end;
end;

if (hh3>(1-hrozptyl)*hstred & hh3<(1+hrozptyl)*hstred)
if (edu(I+8,2)*pokles>edu(I+8,1) & edu(I+8,2)>0)
START(I,3)=START(I,3)+1;
end;
end;
end;
end;
end;
for I=1:4
if ((START(I,1)+START(I,2)+START(I,3))>3)
navrh=I
START
eduf2=zeros(12,1);
for I=1:12
if (eduf2(I)*0.95>edu(I,1) & eduf2(I)>0)
eduf2(I)=eduf2(I)*0.95;
else
eduf2(I)=edu(I,1);
end;
end;
end;
Hpor(1)=mean([eduf2(1) eduf2(5) eduf2(9)]);
Hpor(2)=mean([eduf2(2) eduf2(6) eduf2(10)]);
Hpor(3)=mean([eduf2(3) eduf2(7) eduf2(11)]);
Hpor(4)=mean([eduf2(4) eduf2(8) eduf2(12)]);
Hpor(6)=10;
if (((navrh==1) | (navrh==5) | (navrh==9)) & Hpor(1)<Hpor(porucha)*0.95)
porucha=1;
navrh=0;
pocitadlo=3;
end;

```

```

    if (((navrh==2) / (navrh==6) / (navrh==10)) & Hpor(2)<Hpor(porucha)*0.95)
        porucha=2;
        navrh=0;
        pocitadlo=3;
    end;
    if (((navrh==3) / (navrh==7) / (navrh==11)) & Hpor(3)<Hpor(porucha)*0.95)
        porucha=3;
        navrh=0;
        pocitadlo=3;
    end;
    if (((navrh==4) / (navrh==8) / (navrh==12)) & Hpor(4)<Hpor(porucha)*0.95) /
navrh==porucha)
        porucha=4;
        navrh=0;
        pocitadlo=3;
    end;
end;
end;

```

Blok zapojení regulátoru – předchozí blok programu zjistil, která porucha vstoupila do soustavy, což je první ze dvou kroků. V druhém kroku je třeba zjistit, který ze tří regulátorů, jenž jsou naladěny pro danou poruchu, zapojit. Toto je rozhodnuto na základě hodnot filtru F2. Do regulace je zapojen ten regulátor, který má v daném kroku nejmenší hodnotu tohoto filtru, přičemž aktuálně zapojený regulátor je zvýhodněn koeficientem 0,95.

```

switch porucha
case 1
    [HHH indx]=min([(eduf2(1)*penal(1)) (eduf2(5)*penal(5)) (eduf2(9)*penal(9))]);
    SwContr=(indx-1)*4+1;
    penal=ones(12,1);
    penal((indx-1)*4+1)=.95;
case 2
    [HHH indx]=min([(eduf2(2)*penal(2)) (eduf2(6)*penal(6)) (eduf2(10)*penal(10))]);
    SwContr=(indx-1)*4+2;
    penal=ones(12,1);
    penal((indx-1)*4+2)=.95;
case 3
    [HHH indx]=min([(eduf2(3)*penal(3)) (eduf2(7)*penal(7)) (eduf2(11)*penal(11))]);
    SwContr=(indx-1)*4+3;
    penal=ones(12,1);
    penal((indx-1)*4+3)=.95;
case 4
    [HHH indx]=min([(eduf2(4)*penal(4)) (eduf2(8)*penal(8)) (eduf2(12)*penal(12))]);
    SwContr=indx*4;
    penal=ones(12,1);
    penal(indx*4)=.95;
case 5
    [HHH indx]=min([(eduf2(2)*penal(2)) (eduf2(6)*penal(6)) (eduf2(10)*penal(10))]);
    SwContr=(indx-1)*4+2;

```

```

    penal=ones(12,1);
    penal((indx-1)*4+2)=.95;
end;

```

Blok detekce poruch – je zde proto, aby když vstoupí do soustavy nová porucha, byl zapojen nejrobustnější regulátor, kterým je regulátor OUT, a to jen po dobu, než je tato nová porucha identifikována. Tento blok využívá podobného filtru jako je F2, ale koeficient je zde volen 0,8.

```

if sIN*4<(mean([eduf1(2) eduf1(6) eduf1(10)])-sIN)
    porucha=4;
    SwContr=8;
    pocitadlo=0;
    blok=0;
end;
if sIN<mean([eduf1(2) eduf1(6) eduf1(10)])
    sIN=mean([eduf1(2) eduf1(6) eduf1(10)]);
else
    sIN=.8*sIN;
end;

```

Blok SYS – slouží k exportu námi zvolených proměnných z S-funkce do simulačního schématu, kde je možno s nimi dále pracovat. Jediné pravidlo, které je nutné dodržet je, že všechny proměnné musí být řádkové vektory.

```

sys=[SwContr ein earx earmax eout porucha eduf2' sIN];

```

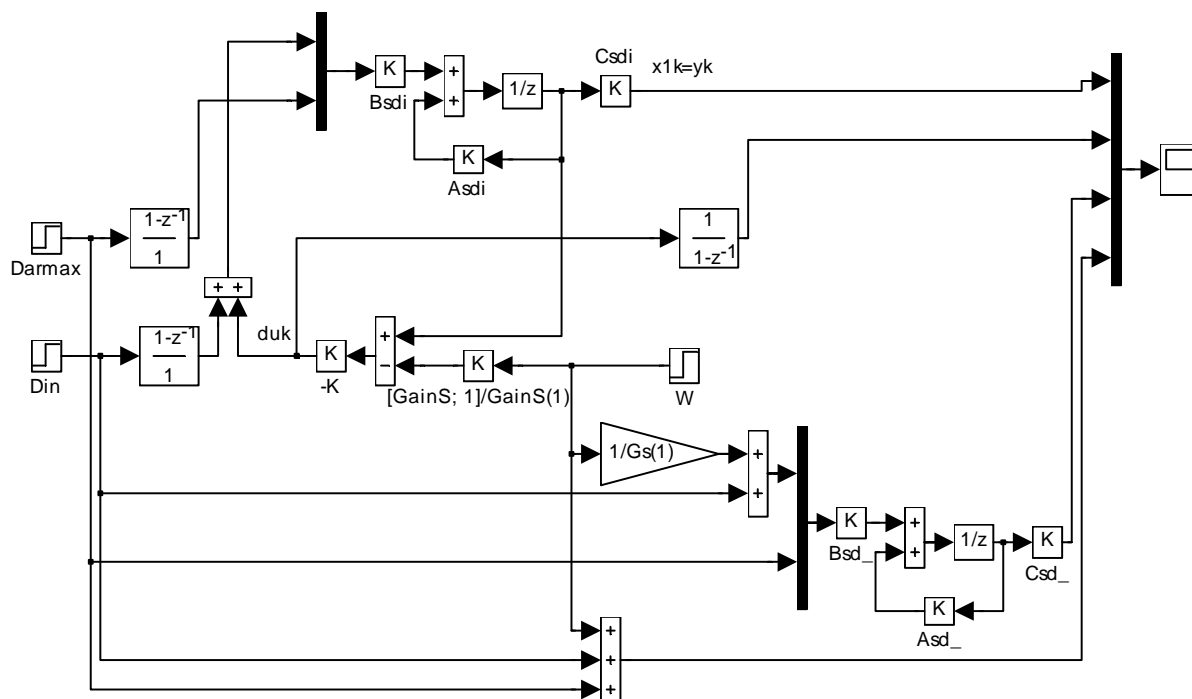
Popis ovládání simulačního programu

Po spuštění je nejprve třeba zadat koeficient časové afinity soustavy, kterou chceme zařadit do simulace. Dále se program ptá na počet poruch, které mají vstoupit do simulace. Maximální počet poruch je pět, přičemž každá porucha může být použita pouze jednou. Poté musí být vložen typ poruchy a čas kdy se spustit. Dále je možno nakonfigurovat typ a rozptyl šumových signálů. Pokud není žádoucí, aby nějaký šum vstupoval během simulace, stačí stisknout klávesu ENTER bez zadání hodnoty. Po „odentrování“ posledního dotazu se spustí simulace, která trvá podle toho jak dlouhý simulační čas byl zvolen.

Pro kontrolu funkce a správnosti přepínání jsou do simulačního schéma zapojeny osciloskopy.

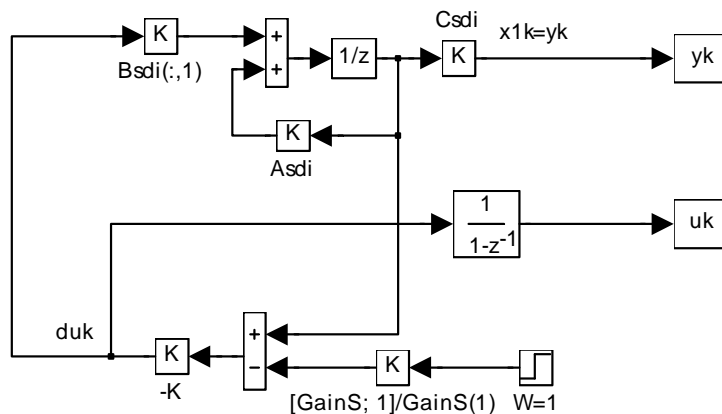
Optimalizace matice Q

Při sestavování kritéria minimální kvadratické regulační plochy se pro penalizaci stavových veličin používá matice Q. Parametry této matice je možné optimalizovat pomocí další simulace.



Obr.7.7 Schéma pro nastavení optimálních parametrů matice Q

A to s využitím jednoduchého schématu, viz. Obr.7.8, kde je zapojen pouze systém a stavový regulátor. Simulace potom probíhá při vstupu pouze jediné poruchy a to změny žádané veličiny, podle schématu viz.Obr.7.7.



Obr.7.8 Simulační schéma pro optimalizaci matice Q

Toho schématu můžeme využít zároveň s funkcí *fminsearch*, která je standardně definovaná v Matlabu a umožňuje multidimensionální nelineární minimalizaci zvolené funkce.

V diplomové práci byl využit tento tvar funkce:

```
x = fminsearch(FUN, X0, OPTIONS)
```

, kde FUN. odkazuje na simulační schéma

X_0 počáteční vektor

OPTIONS . . vlastní nastavení minimalizace

Nastavení parametrů simulace se provádí pomocí další funkce – *optimset* a ta má následující syntaxi:

OPTIONS = OPTIMSET(parametr1, hodnota1, parametr2, hodnota2, . . .)

Byly nastaveny následující parametry:

MaxIter – 100 maximální počet iterací
 MaxFunEvals – 10000000 maximální povolený počet vyhodnocení funkce
 TolX - 1e-22 přerušovací tolerance pro X
 TolFun - 1e-22 přerušovací tolerance pro funkci
 Display – iter nastavení zobrazování výsledků minimalizace

Vlastní minimalizační funkce – m-file

```
function f=OptReg17(x);
Q=zeros(Ord+1); Q(1,1)=x(1); Q(2,2)=x(2); Q(1,2)=x(3); Q(2,1)=x(3);
if min(eig(Q))<0 f=inf;
    disp('*****');
    break;
end;
K=dlqr(Asdi ,Bsdi(:,1),Q,R);
sim('sysReg17',Tmax);
wu=max(uk)-2/Gs(1);
yk(45:end)=[ ];
wy=max(yk)-1.02;
f=wu*wu+.1*wy*wy;
```

Popis průběhu minimalizace pomocí funkce *fminsearch*

Nejprve se nastaví počáteční hodnoty matice Q na hodnoty vektoru X. Dále pokud jsou vlastní čísla matice Q menší než nula tak se výpočet v tomto kroku přeruší.

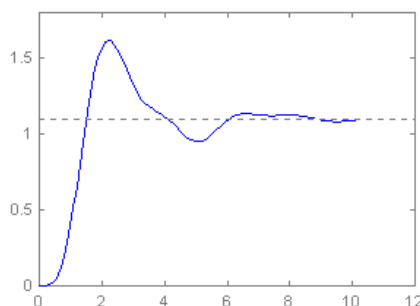
V každém iteračním kroku se počítá matice regulátoru podle kriteria minimální kvadratické regulační plochy, ten je zapojen ve schématu, viz Obr.7.8. Výsledky simulace jsou vráceny do minimalizační funkce. Kde jsou zpracovány jak údaje o akční veličině, tak údaje o regulované veličině. Dále je sestavena funkce f, kterou se snaží tento algoritmus minimalizovat. Vše se opakuje po definovaný počet kroků iterace.

Pomocí definovaných proměnných *wu* a *wy* můžeme ovlivňovat přímo průběh regulované i akční veličiny. Proměnná *wy* určuje maximální tolerovaný překmit výstupní veličiny nad jedničku. A proměnná *wu* představuje maximální možný akční zásah.

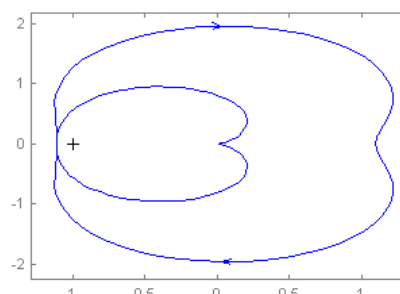
8. Ověření vlastností

Testování při vstupu deterministických poruch - Soustava 1

Pro otestování vlastností naprogramovaného algoritmu bylo použito několika různých soustav. Jako první byl algoritmus testován se soustavou čtvrtého řádu, jejíž charakteristiky jsou na Obr.8.1a, b.



Obr.8.1a Přechodová charakteristika



Obr.8.1b Frekvenční charakteristika

Její přenos je roven

$$F(s) = \frac{19.55}{s^4 + 1.8s^3 + 11.9s^2 + 10.71s + 17.17} \quad (47)$$

póly jsou: $-0.4 \pm 3i$ a $-0.5 \pm 1.3i$

Korekční matice pro jednotlivé typy estimátorů mají tvary:

$$L_{IN} = \begin{bmatrix} 0 & 0 & 0 \\ 1.1646 & 1.1681 & 1.1710 \\ 0.2062 & 0.2066 & 0.2070 \\ 0.0100 & 0.0100 & 0.0100 \\ 0 & 0 & 0 \end{bmatrix}$$

L_{ARX} = nulová matice

(48)

$$L_{ARMAX} = \begin{bmatrix} 0 & 0 & 0 \\ -1.8862 & -1.9822 & -2.0628 \\ 1.1790 & 1.3048 & 1.4148 \\ -0.2436 & -0.2849 & -0.3224 \\ 0 & 0 & 0 \end{bmatrix}$$

L_{OUT} = nulová matice 4 x 3 a navíc řádek jedniček pro sepnutí zpětné vazby

Pro srovnání výsledků je použit průběh výstupní veličiny soustavy regulované regulátorem s proměnnou strukturou, který by vyvinut na TU v Liberci. Tento algoritmus pro přepínání též využívá absolutních hodnot chyb jednotlivých estimátorů, postup vyhodnocení je však odlišný. Tento algoritmus budeme dále označovat jako A2.

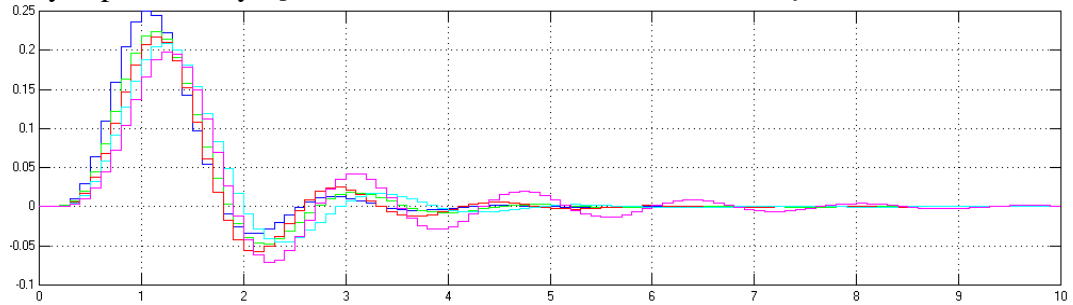
Pro následující grafy byl koeficient afinity pro seřízení regulátorů nastaven na 0.9 – 1 – 1.1 a koeficient afinity regulované soustavy byl měněn od 0,86 do 1,14.

Porucha IN

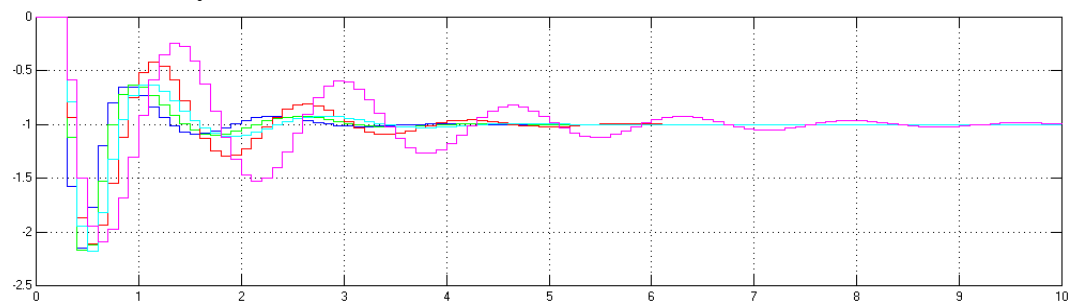
Koeficient afinity regulované soustavy

— 0.86 — 0.95 — 1 — 1.05 — 1.14

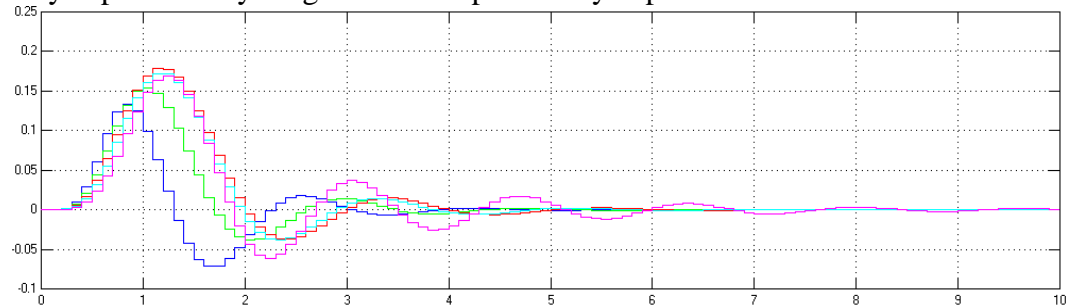
průběh výstupní veličiny - porucha d_{IN} vzorkování = 0.1s , koeficient afinity = 0.86, 0.95, 1, 1.05, 1.14



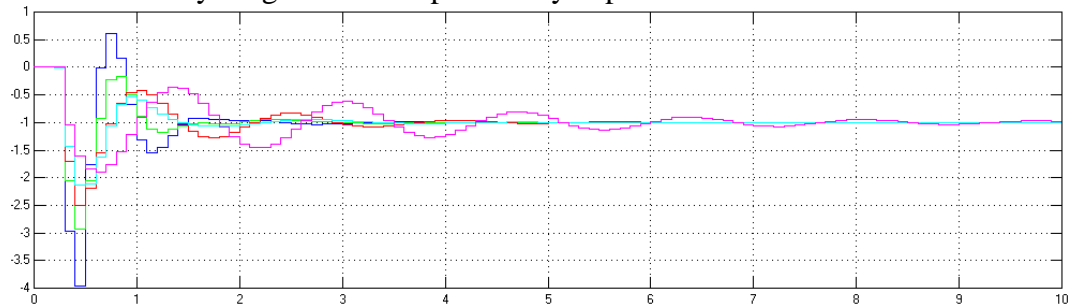
průběh akční veličiny - porucha d_{IN} vzorkování = 0.1s , koeficient afinity = 0.86, 0.95, 1, 1.05, 1.14



průběh výstupní veličiny - algoritmus A2 při shodných podmínkách



průběh akční veličiny - algoritmus A2 při shodných podmínkách

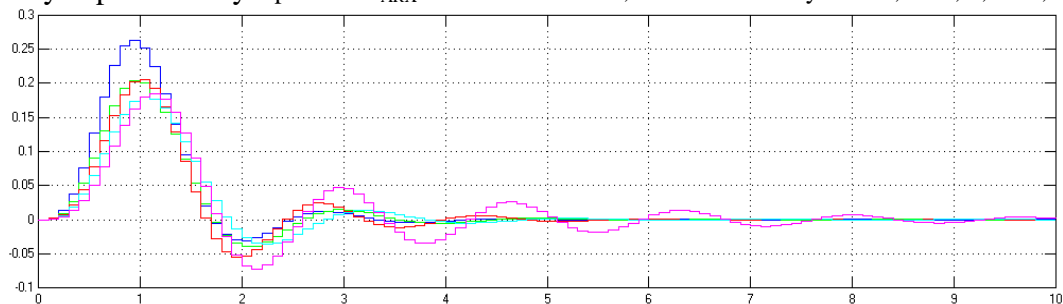


Porucha ARX

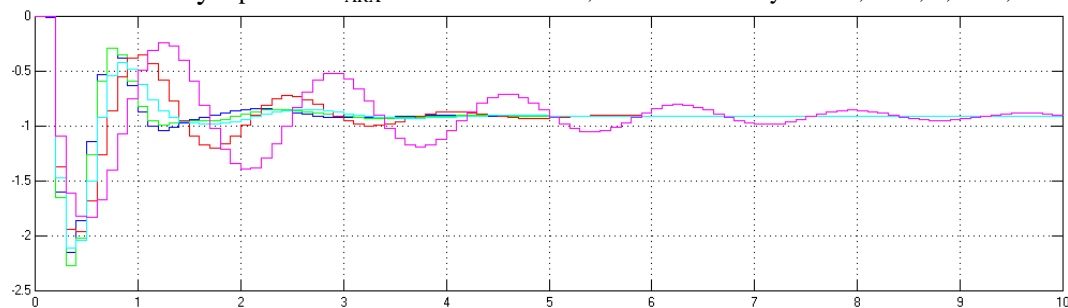
Koeficient afinity regulované soustavy

— 0.86 — 0.95 — 1 — 1.05 — 1.14

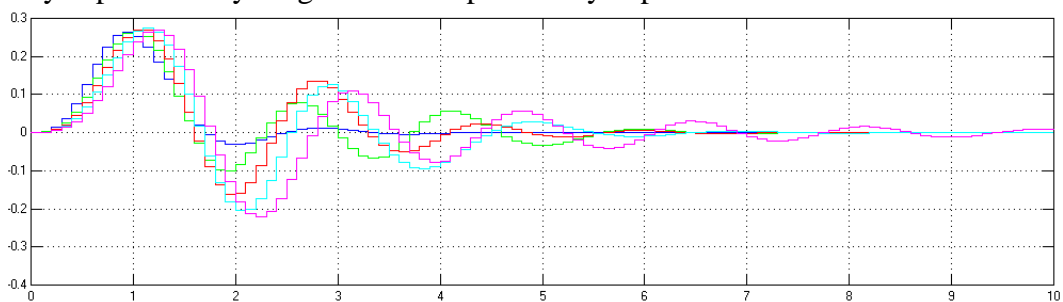
průběh výstupní veličiny - porucha d_{ARX} vzorkování = 0.1s , koeficient afinity = 0.86, 0.95, 1, 1.05, 1.14



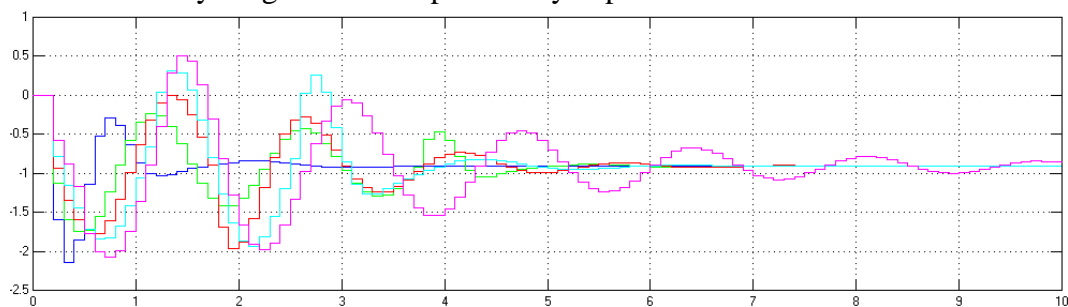
průběh akční veličiny - porucha d_{ARX} vzorkování = 0.1s , koeficient afinity = 0.86, 0.95, 1, 1.05, 1.14



průběh výstupní veličiny - algoritmus A2 při shodných podmínkách



průběh akční veličiny - algoritmus A2 při shodných podmínkách

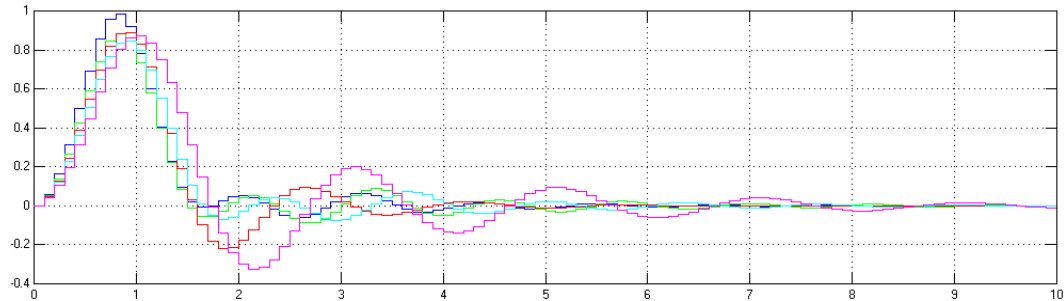


Porucha ARMAX

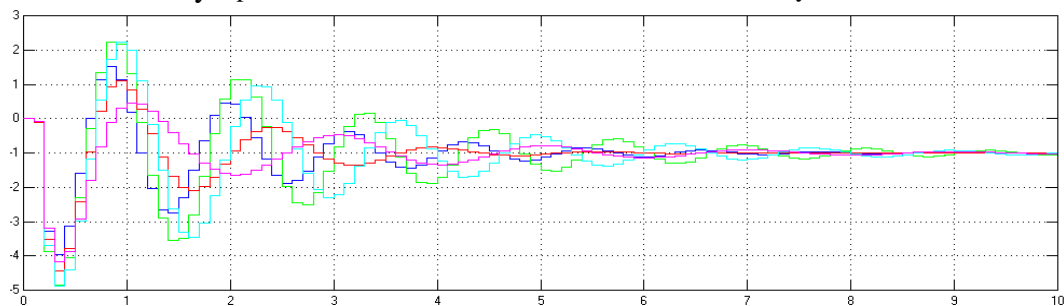
Koeficient afinity regulované soustavy

— 0.86 — 0.95 — 1 — 1.05 — 1.14

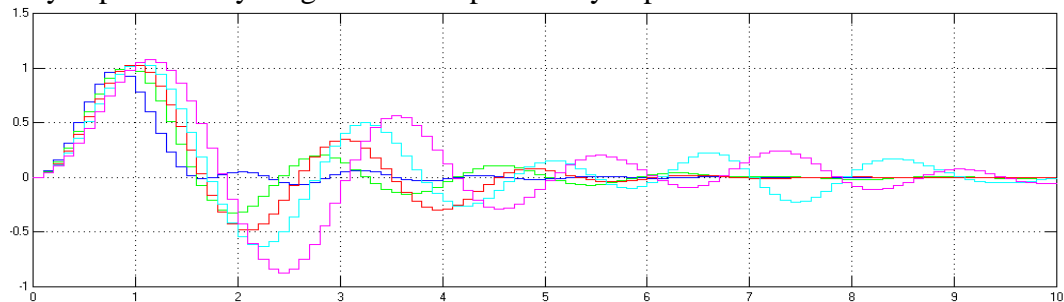
průběh výstupní veličiny - porucha d_{ARMAX} vzorkování = 0.1s , koeficient afinity = 0.86, 0.95, 1, 1.05, 1.14



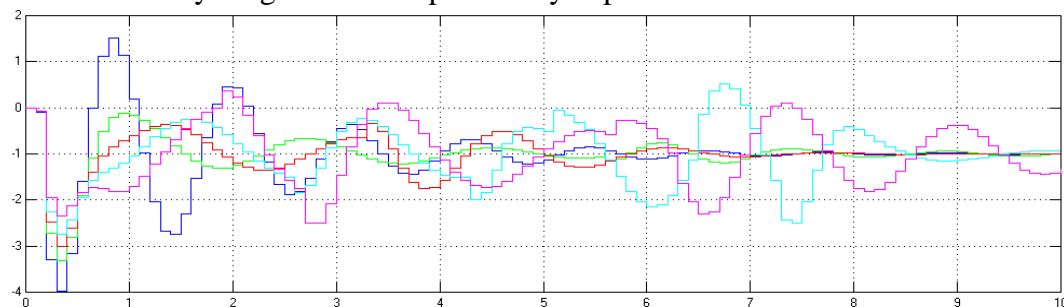
průběh akční veličiny - porucha d_{ARMAX} vzorkování = 0.1s , koeficient afinity = 0.86, 0.95, 1, 1.05, 1.14



průběh výstupní veličiny - algoritmus A2 při shodných podmínkách



průběh akční veličiny - algoritmus A2 při shodných podmínkách

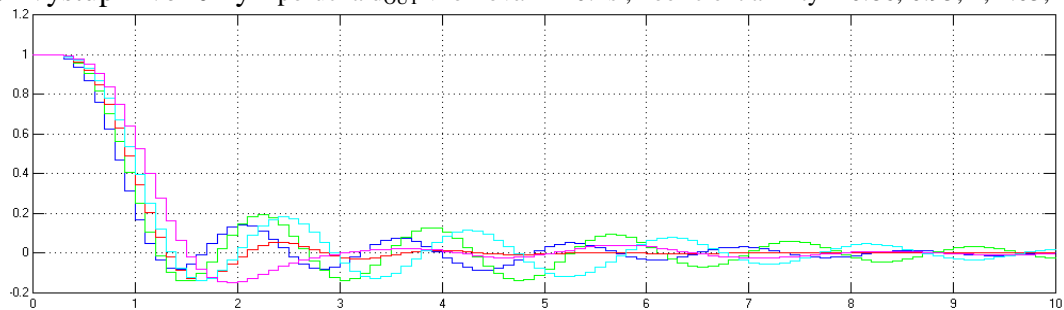


Porucha OUT

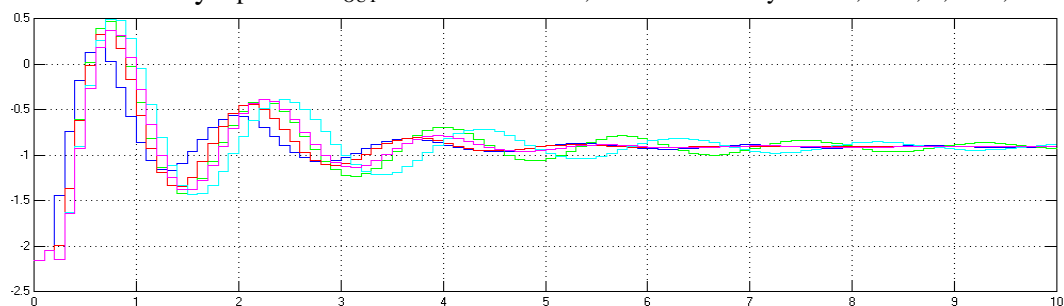
Koeficient afinity regulované soustavy

— 0.86 — 0.95 — 1 — 1.05 — 1.14

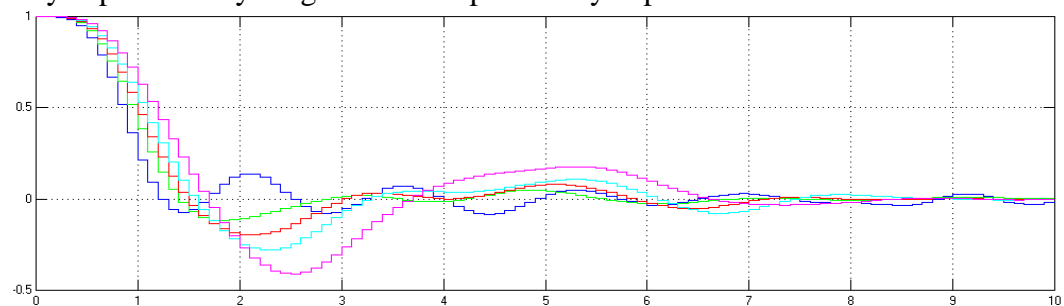
průběh výstupní veličiny - porucha d_{OUT} vzorkování = 0.1s , koeficient afinity = 0.86, 0.95, 1, 1.05, 1.14



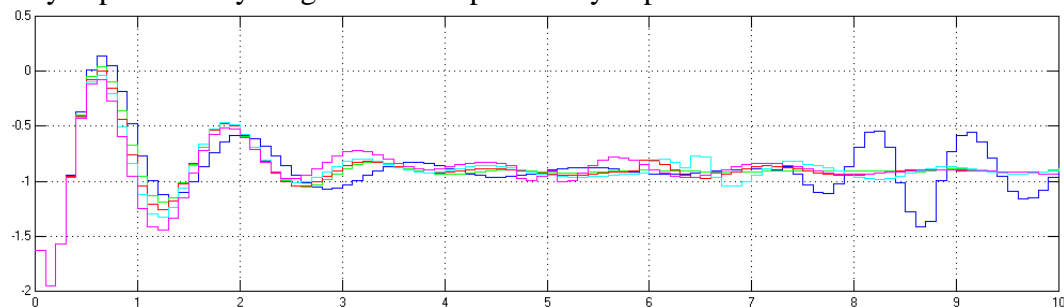
průběh akční veličiny - porucha d_{OUT} vzorkování = 0.1s , koeficient afinity = 0.86, 0.95, 1, 1.05, 1.14



průběh výstupní veličiny - algoritmus A2 při shodných podmínkách

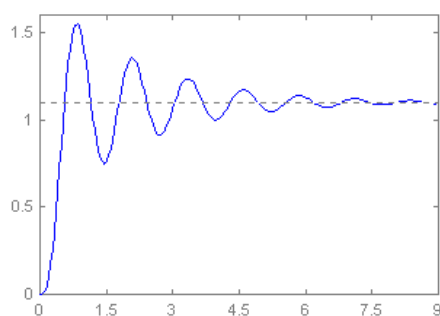


průběh výstupní veličiny - algoritmus A2 při shodných podmínkách

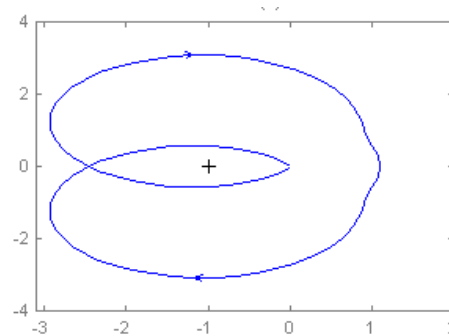


Testování při vstupu deterministických poruch - Soustava 2

Dále byl algoritmus testován na soustavě třetího řádu, která je podstatně kmitavější. Charakteristiky viz. Obr.8.2a, b.



Obr.8.2a Přechodová charakteristika



Obr.8.2b Frekvenční charakteristika

Její přenos je roven

$$F(s) = \frac{111.1}{s^3 + 5s^2 + 29.25s + 101} \quad (49)$$

kořeny jsou: $-0.5 \pm 5i$ a -4

Korekční matice pro jednotlivé typy estimátorů mají tvary:

$$L_{IN} = \begin{bmatrix} 0 & 0 & 0 \\ 0.5489 & 0.5464 & 0.5445 \\ 0.0739 & 0.0735 & 0.0732 \\ 0 & 0 & 0 \end{bmatrix}$$

L_{ARX} = nulová matice

$$L_{ARMAX} = \begin{bmatrix} 0 & 0 & 0 \\ -0.4751 & -0.5412 & -0.5915 \\ 0.0578 & 0.0860 & 0.1084 \\ 0 & 0 & 0 \end{bmatrix}$$

L_{OUT} = nulová matice 3 x 3 a navíc řádek jedniček pro sepnutí zpětné vazby

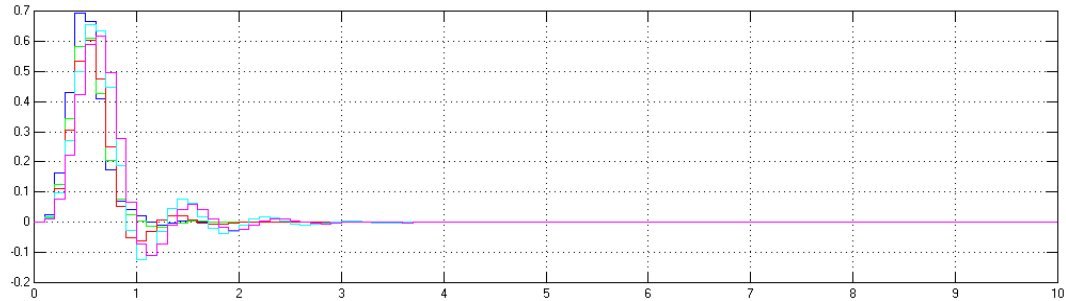
Pro následující grafy byl koeficient afinity pro seřízení regulátorů nastaven na 0.9 – 1 – 1.1

Porucha IN

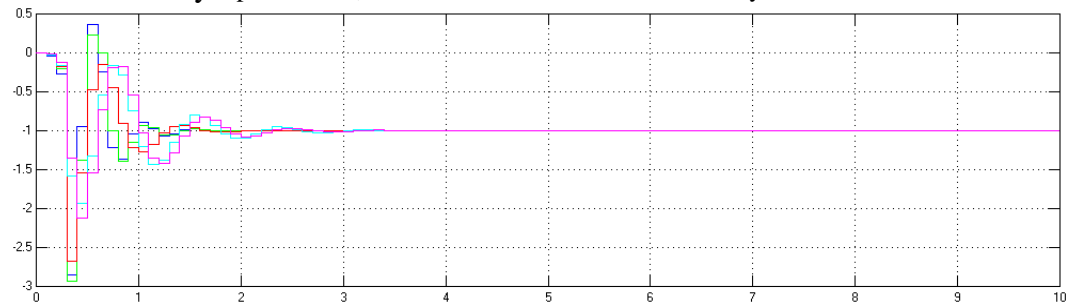
Koeficient afinity regulované soustavy

— 0.86 — 0.95 — 1 — 1.05 — 1.14

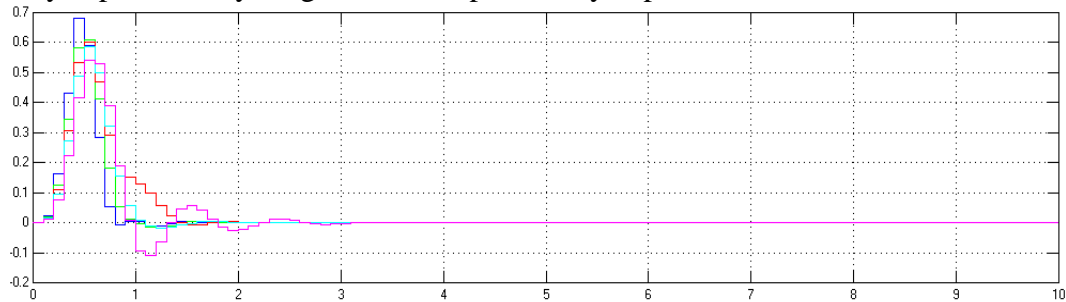
průběh výstupní veličiny - porucha d_{IN} vzorkování = 0.1s , koeficient afinity = 0.86, 0.95, 1, 1.05, 1.14



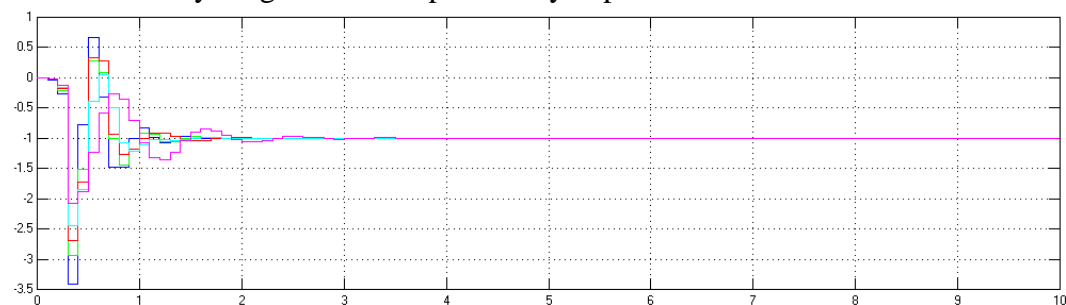
průběh akční veličiny - porucha d_{IN} vzorkování = 0.1s , koeficient afinity = 0.86, 0.95, 1, 1.05, 1.14



průběh výstupní veličiny - algoritmus A2 při shodných podmínkách



průběh akční veličiny - algoritmus A2 při shodných podmínkách

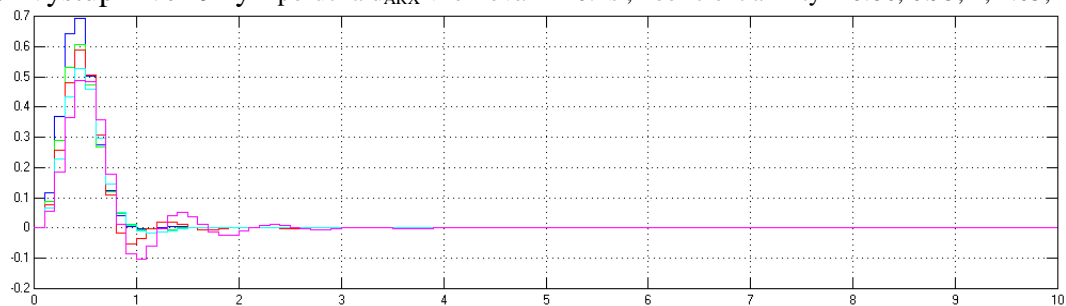


Porucha ARX

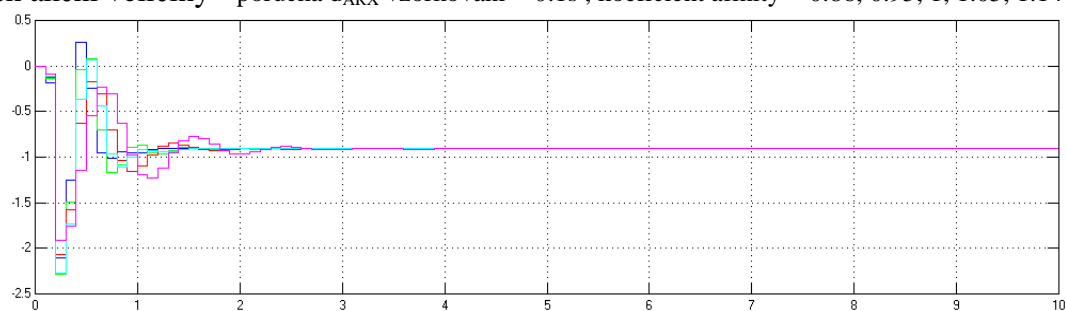
Koeficient afinity regulované soustavy

— 0.86 — 0.95 — 1 — 1.05 — 1.14

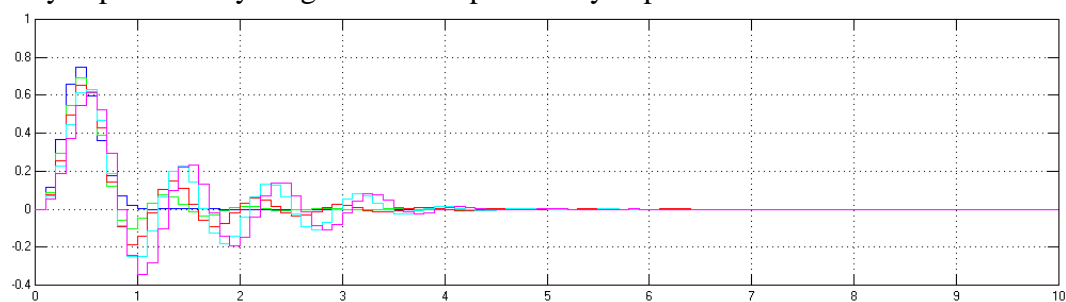
průběh výstupní veličiny - porucha d_{ARX} vzorkování = 0.1s , koeficient afinity = 0.86, 0.95, 1, 1.05, 1.14



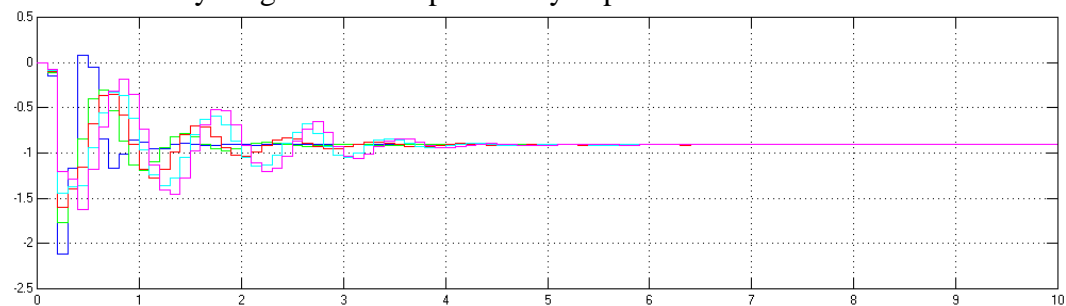
průběh akční veličiny - porucha d_{ARX} vzorkování = 0.1s , koeficient afinity = 0.86, 0.95, 1, 1.05, 1.14



průběh výstupní veličiny - algoritmus A2 při shodných podmínkách



průběh akční veličiny - algoritmus A2 při shodných podmínkách

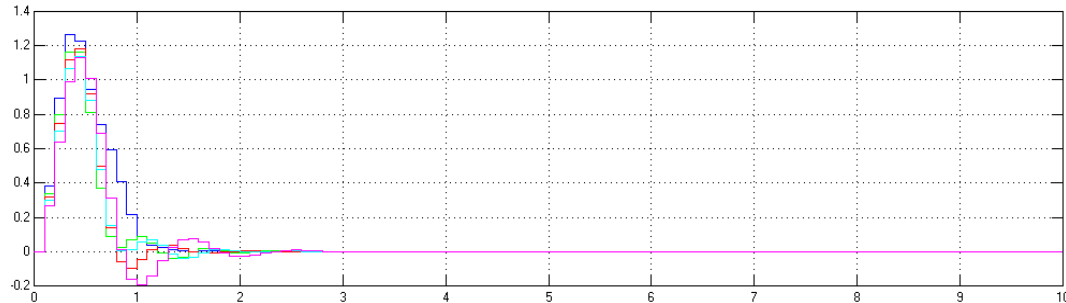


Porucha ARMAX

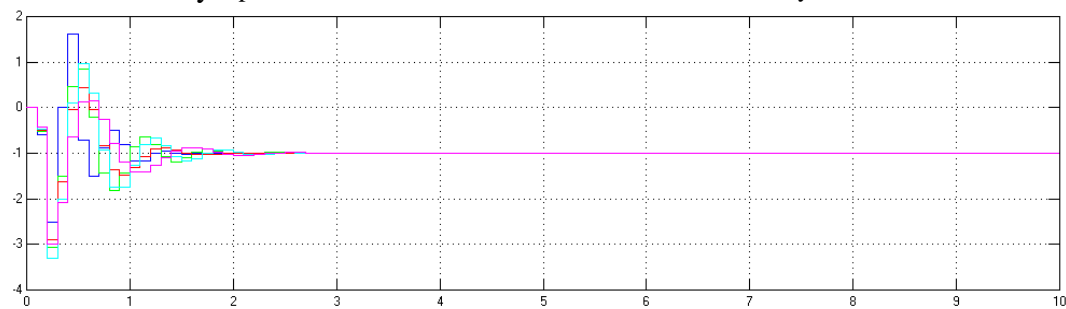
Koeficient afinity regulované soustavy

— 0.86 — 0.95 — 1 — 1.05 — 1.14

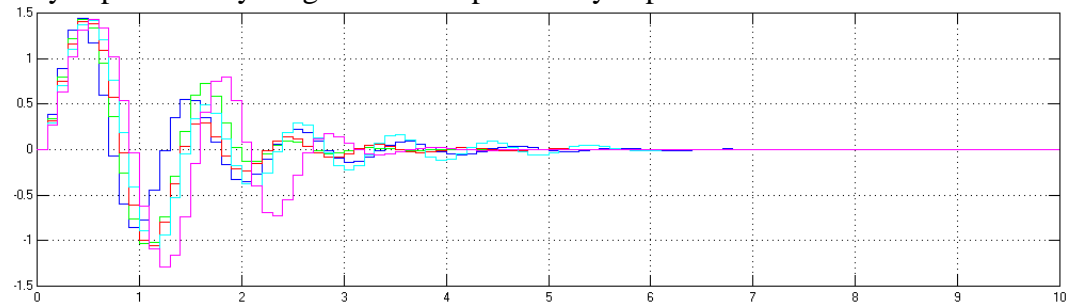
průběh výstupní veličiny - porucha d_{ARMAX} vzorkování = 0.1s , koeficient afinity = 0.86, 0.95, 1, 1.05, 1.14



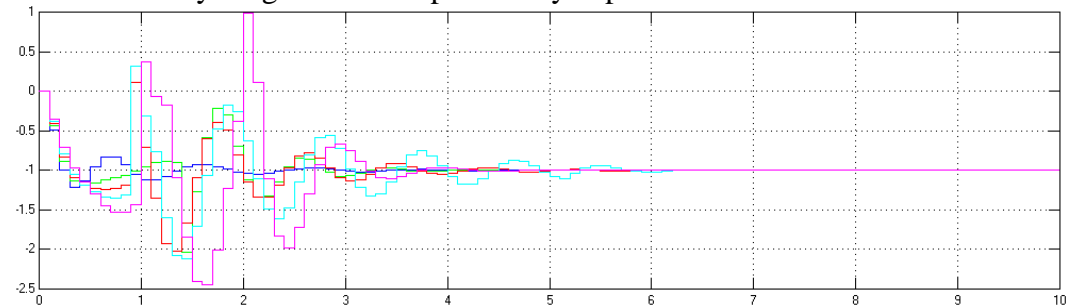
průběh akční veličiny - porucha d_{ARMAX} vzorkování = 0.1s , koeficient afinity = 0.86, 0.95, 1, 1.05, 1.14



průběh výstupní veličiny - algoritmus A2 při shodných podmínkách



průběh akční veličiny - algoritmus A2 při shodných podmínkách

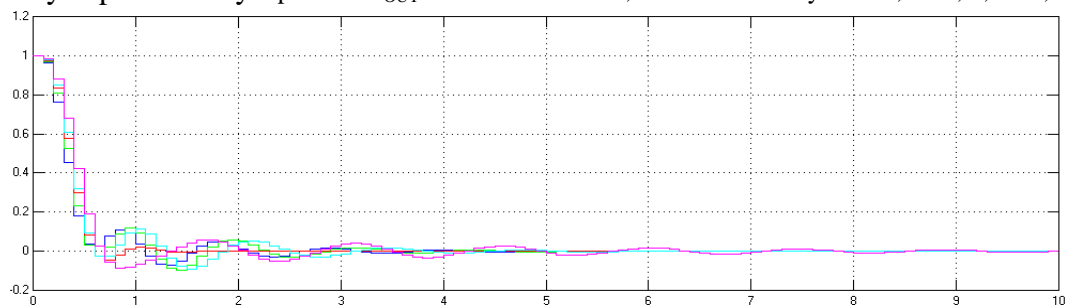


Porucha OUT

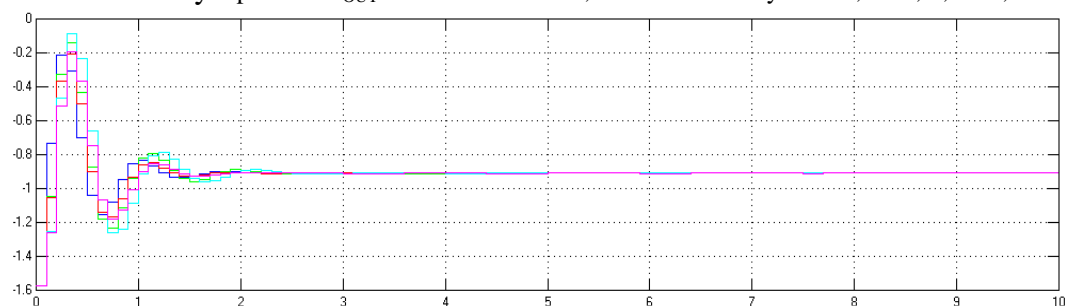
Koeficient afinity regulované soustavy

— 0.86 — 0.95 — 1 — 1.05 — 1.14

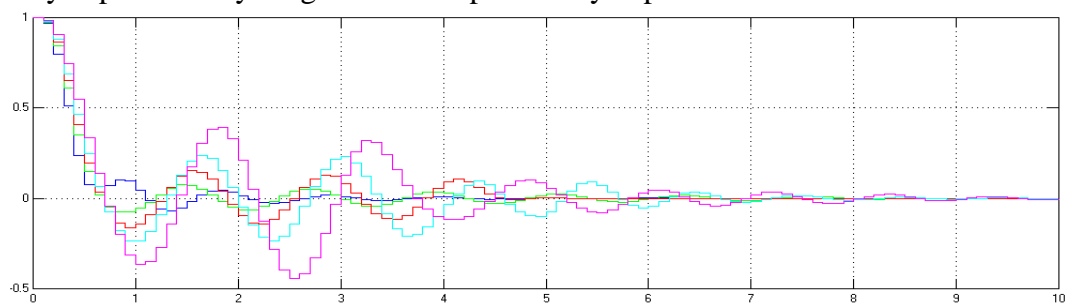
průběh výstupní veličiny - porucha d_{OUT} vzorkování = 0.1s , koeficient afinity = 0.86, 0.95, 1, 1.05, 1.14



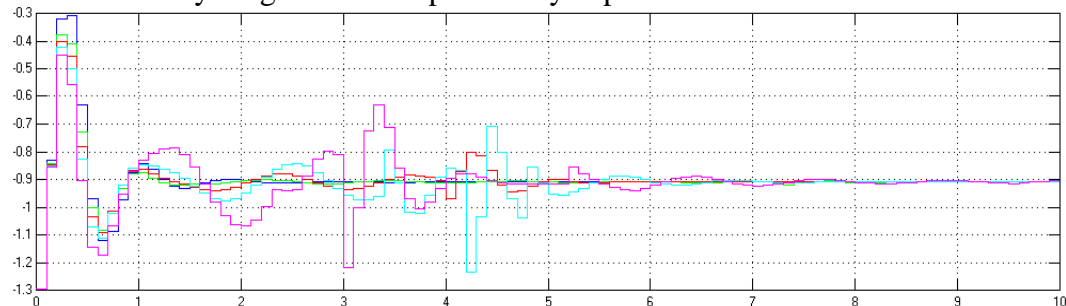
průběh akční veličiny - porucha d_{OUT} vzorkování = 0.1s , koeficient afinity = 0.86, 0.95, 1, 1.05, 1.14



průběh výstupní veličiny - algoritmus A2 při shodných podmínkách



průběh akční veličiny - algoritmus A2 při shodných podmínkách



Testování s „parazitní kapacitou“ I

Pro další testování byla použita tzv. parazitní kapacita, kterou jsme připojili do serie za regulovanou soustavu, která byla použita již při prvním testování. Její přenos je

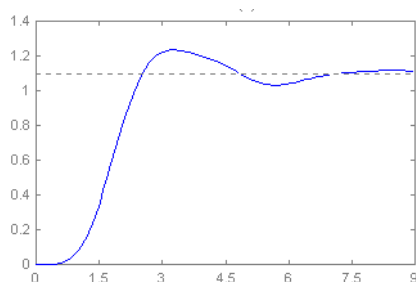
$$F_p = \frac{1}{10 \cdot T_s + 1} \quad (50)$$

,kde T_s je perioda vzorkování ($T_s = 0.1$).

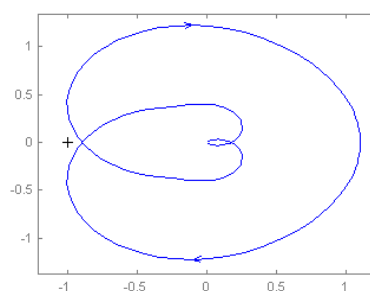
Výsledný přenos obou soustav je potom vyjádřen

$$F(s) = \frac{19.55}{s^5 + 2.8s^4 + 13.7s^3 + 22.61s^2 + 28.48s + 17.17} \quad (51)$$

Charakteristiky:



Obr.8.3a Přechodová charakteristika



Obr.8.3b Frekvenční charakteristika

Při nastavování jednotlivých regulátorů již nebylo využíváno časové afinity soustavy. Regulátory byly vypočítány podle soustavy, která vznikla tak, že se pól (jmenovatel) parazitní kapacity postupně vydělil třemi koeficienty. Tyto koeficienty byly voleny 0.5 – 1 – 1.5. Čili byly vytvořeny soustavy, které mají časově afinní parazitní kapacity.

Korekční matice mají tvar:

$$L_{IN} = \begin{bmatrix} 0 & 0 & 0 \\ 1.1683 & 1.1683 & 1.1683 \\ 0.2067 & 0.2067 & 0.2067 \\ 0.0100 & 0.0100 & 0.0100 \\ 0.0000 & 0.0000 & 0.0000 \\ 0 & 0 & 0 \end{bmatrix}$$

L_{ARX} = nulová matice

$$L_{ARMAX} = \begin{bmatrix} 0 & 0 & 0 \\ -1.8862 & -1.9820 & -2.0624 \\ 1.1792 & 1.3045 & 1.4141 \\ -0.2437 & -0.2846 & -0.3220 \\ -0.0000 & -0.0000 & -0.0001 \\ 0 & 0 & 0 \end{bmatrix}$$

L_{OUT} = nulová matice 5 x 3 a navíc řádek jedniček pro sepnutí zpětné vazby

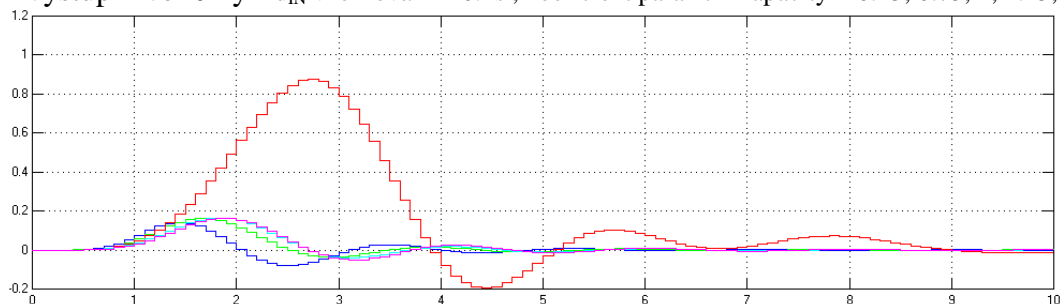
Pro následující testování byl kořen přenosu parazitní kapacity dělen pěti různými konstantami, čímž jsme získali soustavy s odlišným přenosem.

Porucha IN

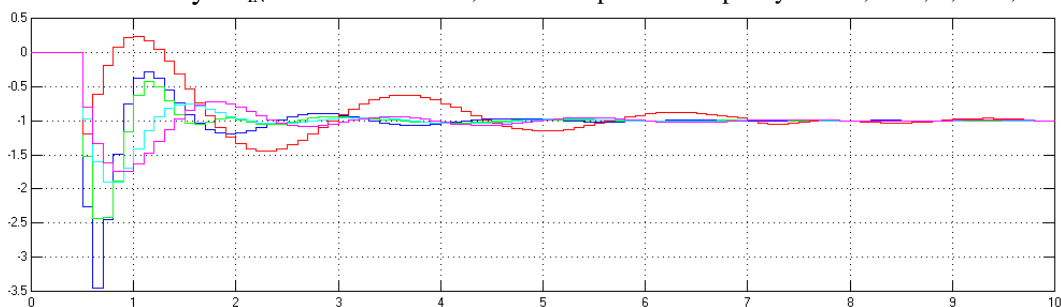
Koeficient parazitní kapacity

— 0.45 — 0.75 — 1 — 1.25 — 1.55

průběh výstupní veličiny - d_{IN} vzorkování = 0.1s , koeficient parazitní kapacity = 0.45, 0.75, 1, 1.25, 1.55

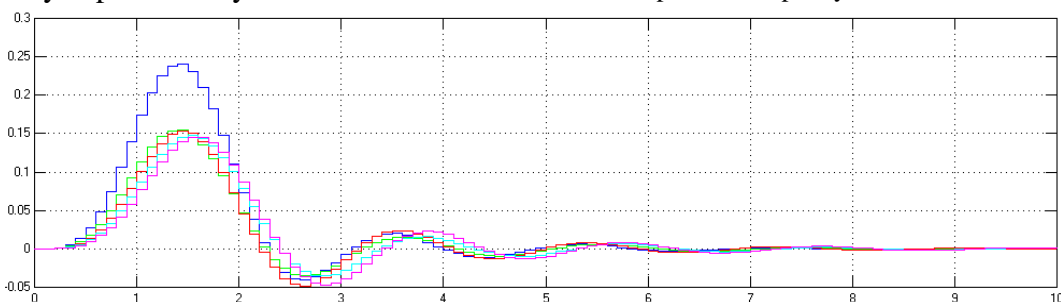


průběh akční veličiny - d_{IN} vzorkování = 0.1s , koeficient parazitní kapacity = 0.45, 0.75, 1, 1.25, 1.55

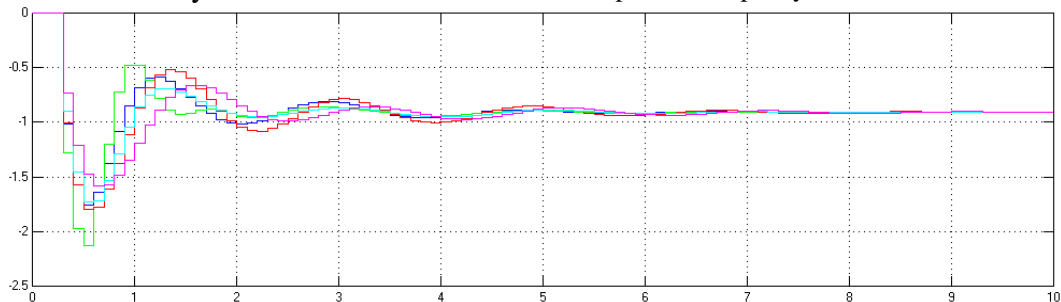


Porucha ARX

průběh výstupní veličiny - d_{ARX} vzorkování = 0.1s , koeficient parazitní kapacity = 0.45, 0.75, 1, 1.25, 1.55



průběh akční veličiny - d_{ARX} vzorkování = 0.1s , koeficient parazitní kapacity = 0.45, 0.75, 1, 1.25, 1.55

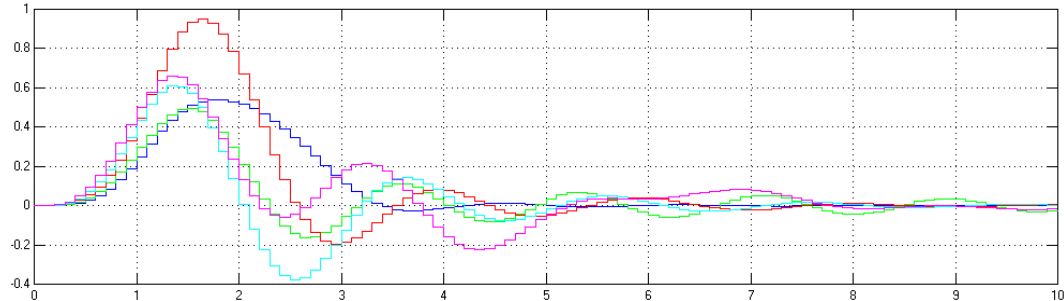


Porucha ARMAX

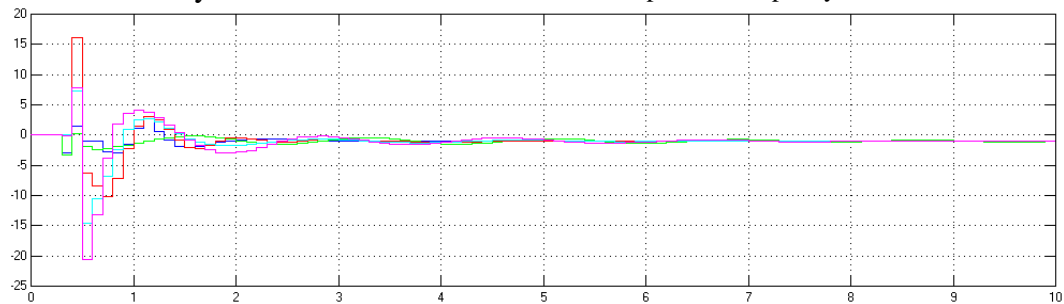
Koeficient parazitní kapacity

— 0.45 — 0.75 — 1 — 1.25 — 1.55

průběh výstupní veličiny - d_{ARMAX} vzorkování = 0.1s , koef. parazitní kapacity = 0.45, 0.75, 1, 1.25, 1.55

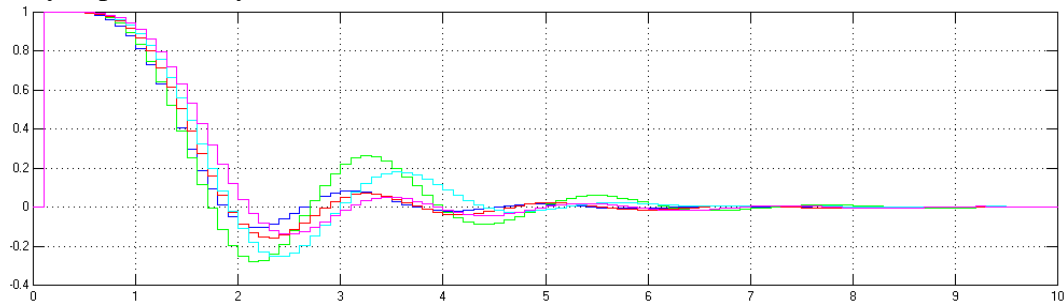


průběh akční veličiny - d_{ARMAX} vzorkování = 0.1s , koeficient parazitní kapacity = 0.45, 0.75, 1, 1.25, 1.55

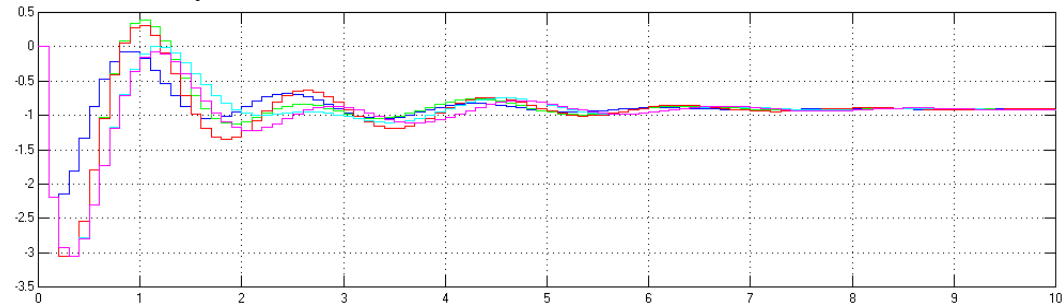


Porucha OUT

průběh výstupní veličiny - d_{OUT} vzorkování = 0.1s , koeficient parazitní kapacity = 0.45, 0.75, 1, 1.25, 1.55



průběh akční veličiny - d_{OUT} vzorkování = 0.1s , koeficient parazitní kapacity = 0.45, 0.75, 1, 1.25, 1.55



Testování s „parazitní kapacitou“ II

Pro testování byla použita „parazitní kapacita“, ale tentokrát nebyla zahrnuta do výpočtu regulátorů a estimátorů. Její přenos je vyjádřen v (47). Přenos parazitní kapacity je

$$F_p = \frac{1}{K \cdot s + 1} \quad (52)$$

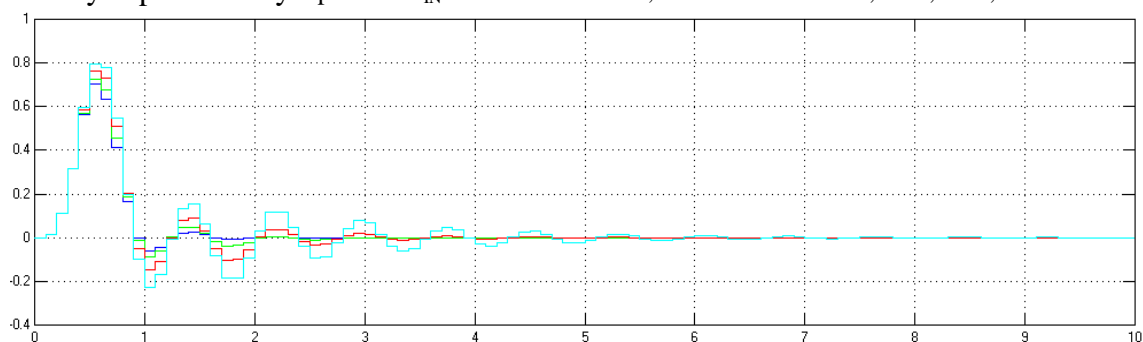
kde K je koeficient, který byl v průběhu testování měněn v rozsahu 0 až 0.08. Bylo též ověřeno, že uzavřený regulační obvod s přepínaným regulátorem je stabilní až do hodnoty K=0,11. Testování bylo prováděno na soustavě, která měla koeficient časové afinity roven 1.

Porucha IN

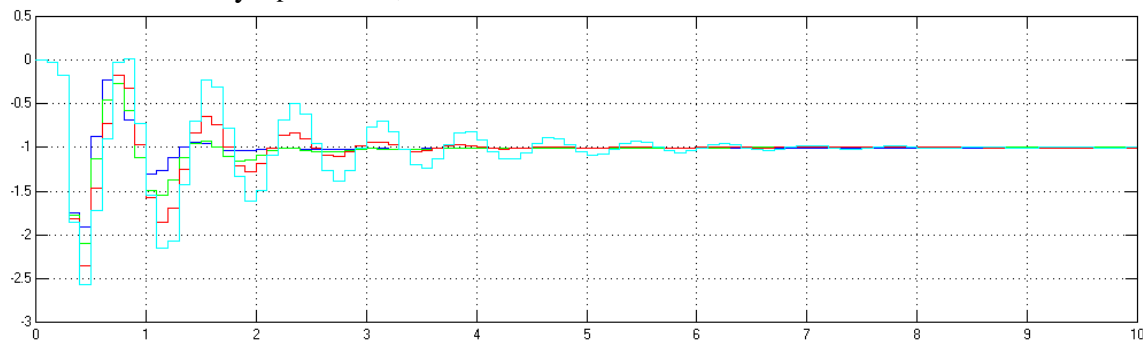
Koeficient K

— 0.02 — 0.04 — 0.06 — 0.08

průběh výstupní veličiny - porucha d_{IN} vzorkování = 0.1s , koeficient K = 0.02, 0.04, 0.06, 0.08



průběh akční veličiny - porucha d_{IN} vzorkování = 0.1s , koeficient K = 0.02, 0.04, 0.06, 0.08

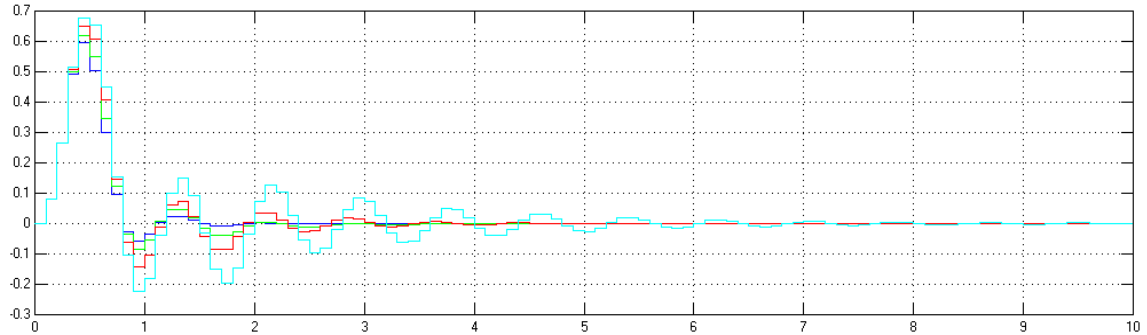


Porucha ARX

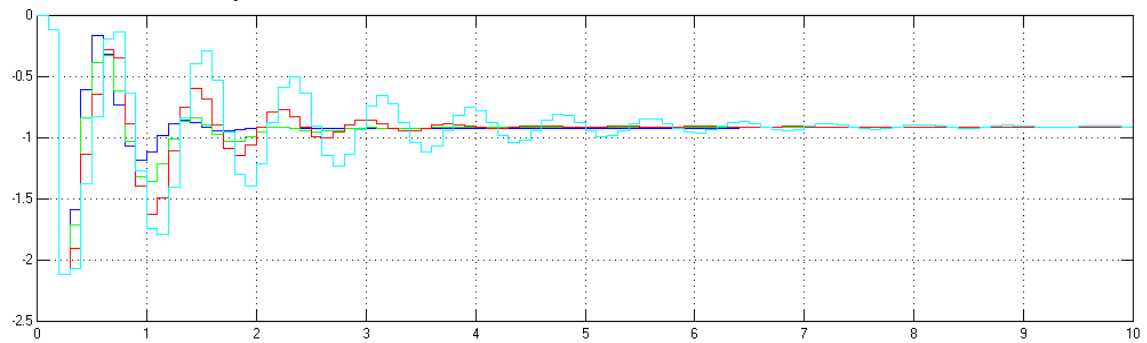
Koeficient K

— 0.02 — 0.04 — 0.06 — 0.08

průběh výstupní veličiny - porucha d_{ARX} vzorkování = 0.1s , koeficient K = 0.02, 0.04, 0.06, 0.08

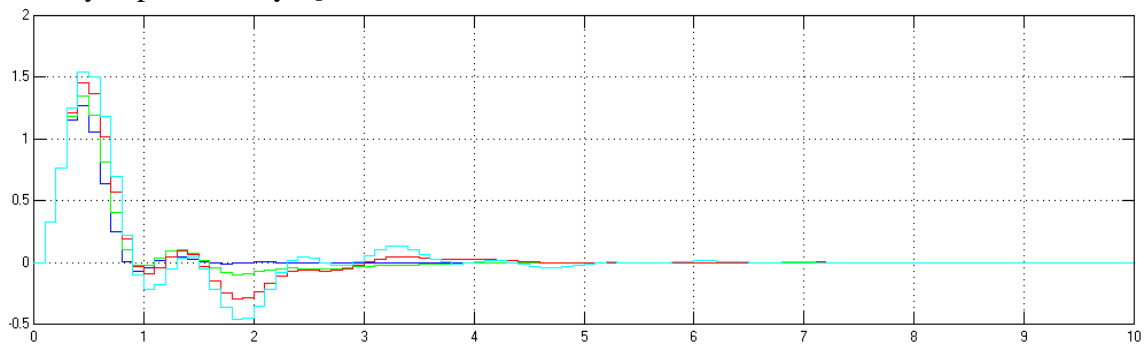


průběh akční veličiny - porucha d_{ARX} vzorkování = 0.1s , koeficient K = 0.02, 0.04, 0.06, 0.08

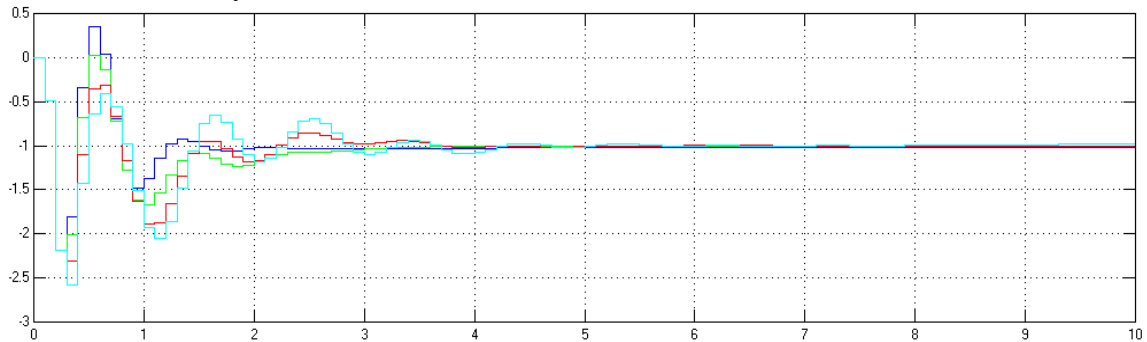


Porucha ARMAX

průběh výstupní veličiny - porucha d_{ARMAX} vzorkování = 0.1s , koeficient K = 0.02, 0.04, 0.06, 0.08



průběh akční veličiny - porucha d_{ARMAX} vzorkování = 0.1s , koeficient K = 0.02, 0.04, 0.06, 0.08

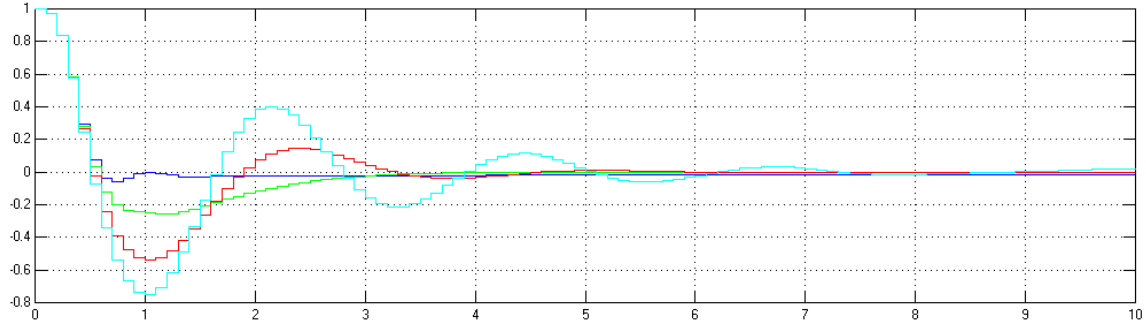


Porucha OUT

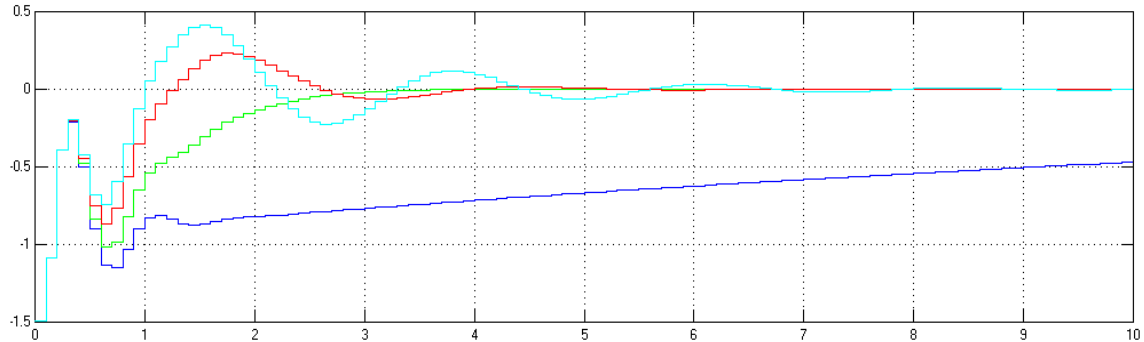
Koeficient K

— 0.02 — 0.04 — 0.06 — 0.08

průběh výstupní veličiny - porucha d_{OUT} vzorkování = 0.1s , koeficient K = 0.02, 0.04, 0.06, 0.08



průběh akční veličiny - porucha d_{OUT} vzorkování = 0.1s , koeficient K = 0.02, 0.04, 0.06, 0.08



Testování při vstupu stochastických poruch

Při testování algoritmu přepínání při vstupu náhodných poruch, byla nejprve použita méně kmitavá soustava viz. soustava 1 (47).

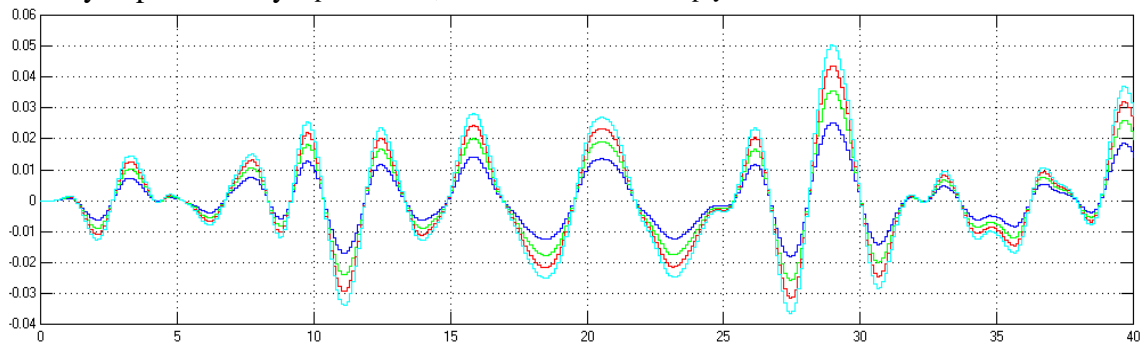
Nastavení korekčních matic jednotlivých estimátorů zůstává stejné jako v (48). Pro testování algoritmu při vstupu stochastických poruch byl do definovaných vstupů soustavy přiváděn náhodný signál o rozptylu 0.05 – 0.1 – 0.15 – 0.2

Porucha IN

Rozptyl náhodného signálu

— 0.05 — 0.01 — 0.15 — 0.2

průběh výstupní veličiny - porucha d_{IN} vzorkování = 0.1s , rozptyl = 0.05, 0.15, 0.15, 0.2

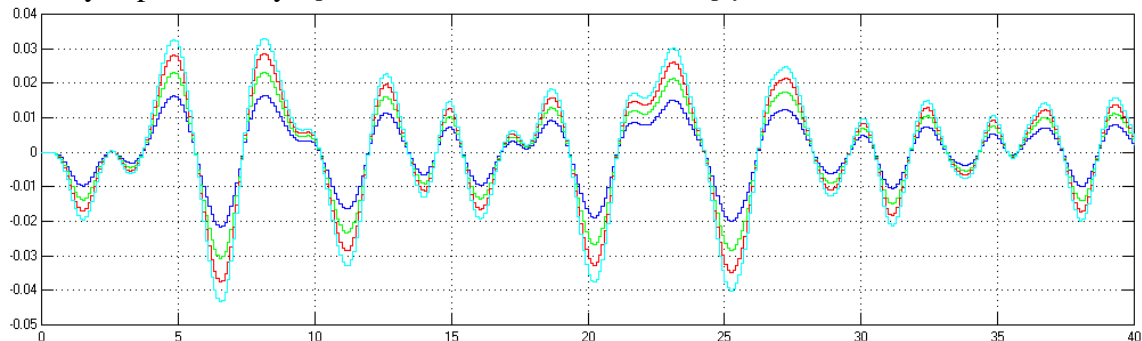


Porucha ARX

Rozptyl náhodného signálu

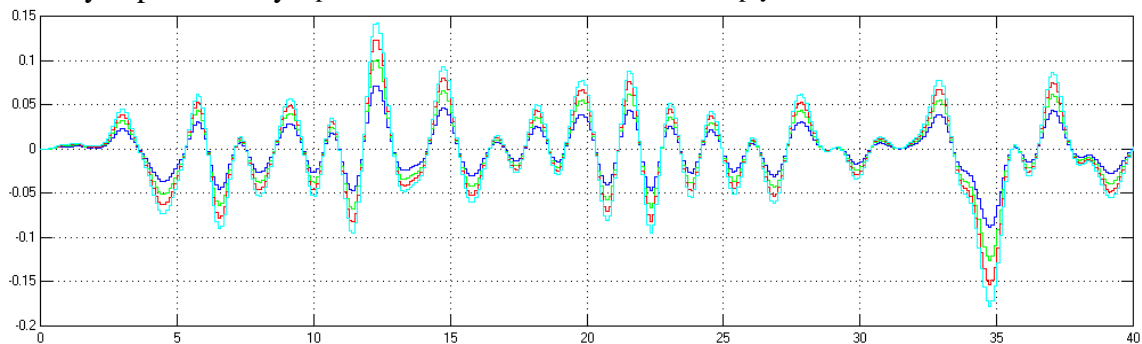
— 0.05 — 0.01 — 0.15 — 0.2

průběh výstupní veličiny - porucha d_{ARX} vzorkování = 0.1s , rozptyl = 0.05, 0.15, 0.15, 0.2



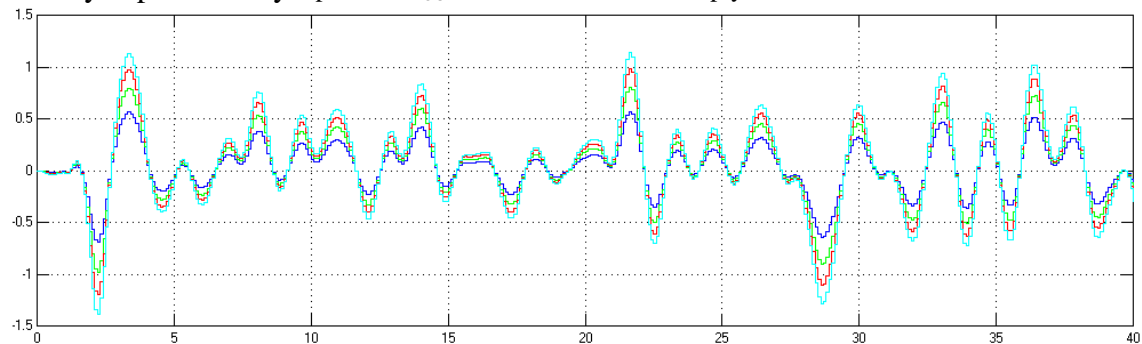
Porucha ARMAX

průběh výstupní veličiny - porucha d_{ARMAX} vzorkování = 0.1s , rozptyl = 0.05, 0.15, 0.15, 0.2



Porucha OUT

průběh výstupní veličiny - porucha d_{OUT} vzorkování = 0.1s , rozptyl = 0.05, 0.15, 0.15, 0.2



9. Hodnocení

Při pohledu na grafy je vidět, že algoritmus vytvořený v této diplomové práci skoro vždy lépe využívá vlastností jednotlivých estimátorů s regulátory než algoritmus A2. Vždy je to způsobeno vhodnějším přepnutím mezi jednotlivými regulátory.

U testování deterministický poruch se soustavou 1 a 2 nenastal při přepínání jednotlivých regulátorů žádný problém, protože jsou estimátory i regulátory nastaveny přesně na soustavu, která je zapojena do obvodu.

Stejně tak je tomu i u testování s „parazitní kapacitou“ I, která časově změní průběh výstupní veličiny. Estimátory a regulátory jsou však na tuto přidanou soustavu upraveny. Respektive jsou nastaveny na soustavy časově afinní s koeficienty afinity $0.9 - 1 - 1.1$, takže pokud v simulaci nastavíme koeficient afinity na hodnotu mezi tato čísla je nutné, aby algoritmus vybral ten estimátor a k němu připojený regulátor, který je svým seřízením neoptimálnější v dané situaci. Toto se podařilo do algoritmu poměrně dobře naprogramovat.

Dále byl algoritmus testován na soustavě 2 při zapojení parazitní kapacity, která však nebyla zahrnuta do výpočtu regulátorů a estimátorů. Jak již bylo zmíněno, tato kapacita se na soustavě projeví tak, že časově prodlouží přechodovou charakteristiku původní soustavy. To je v podstatě podobné, jako vytváření afinních soustav. Algoritmus tedy do regulačního procesu zapojoval regulátory nastavené na „pomalejší“ soustavu, jak jsme předpokládali.

Při testování se stochastickým signálem se algoritmus choval též stabilně. Hodnota amplitudy nepravidelných kmitů na výstupní veličině je úměrná hodnotě rozptylu náhodného signálu, který vstupuje do soustavy.

10. Závěr

Diplomová práce rozšířila poznatky o principech číslicového stavového regulátoru s upraveným estimátorem stavu, který vypočítá bezchybně stav regulované soustavy v případě vstupu dané poruchy do regulačního obvodu.

K vytvoření modelu soustavy a počítačového řízení bylo použito programového produktu firmy Mathworks - MATLAB verze 5.3. Byly použity regulátory navržené metodou minimalizace kvadratického kritéria.

Vypracovaný algoritmus dobře přepínal mezi zvolenými regulátory při testování na soustavě třetího a čtvrtého řádu a vykazoval lepší kvalitu regulace než na začátku práce známý regulátor, který byl označen zkratkou A2.

Příloha A

Osah CD

- Regulátor – vlastní simulační schéma + přepínací funkce + obslužný program
- Simulace – výsledky prováděných simulací, uložené v datových souborech (*.mat)
- Dokumentace – tato diplomová práce v elektronické podobě
- Optimalizace – program pro optimalizaci matice Q v kvadratickém kritériu (m-file)

Regulátor – simulace se spustí zadáním jména VARS do příkazové řádky v Matlabu. Dále se program zeptá na potřebné informace a nasimuluje regulaci.

Simulace – obsahuje několik adresářů s výsledky různých simulací

- kap-s2 – simulace s parazitní kapacitou viz. Testování s „parazitní kapacitou“ II – str.46
- parkap-s1 – simulace s par. kapacitou viz. Testování s „parazitní kapacitou“ I - str. 43
- soustava1 – testování algoritmu na soustavě 1
 - jednapor-01 – při vzorkovací periodě 0.1s
 - jednapor-02 – při vzorkovací periodě 0.2s
 - porsum-01 – při vstupu náhodného signálu na zvolené vstupy a při vzorkování 0.1s
 - porsum-02 – při vstupu náhodného signálu na zvolené vstupy a při vzorkování 0.2s
- soustava2 – stejné druhy simulací jako se soustavou 1

Označení poruch :

p1 – IN p2 – ARX p3 – ARMAX p1 – OUT p5 – změna w

Způsob pojmenování souborů:

- kap-s2 – *porucha-koeficient.mat*
- parkap-s1 – *porucha-koef. par. capacity.mat*
- soustavax - jednapor-0x - *typ poruchy-koeficient afinity.mat*
 - por1sum-0x - *typ poruchy-rozptyl náh. signálu.mat*

Optimalizace

Program optimalizuje matici Q a vypočte její hodnoty pro 3 afinní soustavy. Spouští se zadáním jména RegOpt do příkazové řádky Matlabu.

Seznam použité literatury

- [1] Havlena V., Štecha J. : Moderní teorie řízení.
Editační středisko ČVUT Praha, 1999.
- [2] Modrák O.: Základy analýzy a syntézy ve stavovém prostoru. Studijní materiály Liberec, 2002.
- [3] Hanuš B., Janeček B., Tůma L.:Control System with Variable Structure. Proc. 5th International Scientific – Technical Conference Process Control 2002, Říp 2002, University of Pardubice, 2002, 1-12
- [4] Kalaš D., Jucha I.:MATLAB/Simulink a Control System Toolbox.
STU v Bratislavě, 1996.
- [5] Herajn P., Janeček J.: Návrh stavového regulátoru s ohledem na robustnost uzavřeného regulačního obvodu. Proc. 5th International Scientific – Technical Conference Process Control 2002, Říp 2002, University of Pardubice, 2002, 1-8
- [6] Kotek, Štecha: Teorie automatického řízení spojitých lineárních systémů. Skripta FEL ČVUT, Vydavatelství ČVUT 1978.